

BEST AVAILABLE COPY

PCT/JP 2004/008709

PCT

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

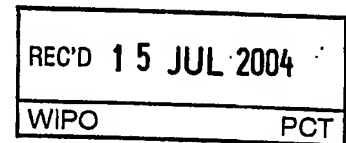
22.06.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 3 年    6 月 2 5 日  
Date of Application:

出 願 番 号                      特 願 2 0 0 3 - 1 8 0 6 5 9  
Application Number:  
[ST. 10/C] :                      [ J P 2 0 0 3 - 1 8 0 6 5 9 ]



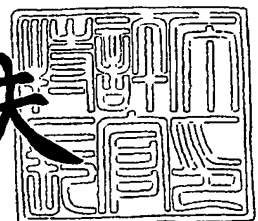
出 願 人                      日 本 電 気 株 式 会 社  
Applicant(s):

PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2 0 0 4 年    4 月 2 2 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号    出証特 2 0 0 4 - 3 0 3 4 5 9 1

【書類名】 特許願  
【整理番号】 34002300  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 9/00  
H01L 27/00

## 【発明者】

【住所又は居所】 東京都港区芝五丁目7番1号  
本電気株式会社内

日

【氏名】 犬尾 武

## 【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

## 【代理人】

【識別番号】 100109313

## 【弁理士】

【氏名又は名称】 机 昌彦

【電話番号】 03-3454-1111

## 【選任した代理人】

【識別番号】 100085268

## 【弁理士】

【氏名又は名称】 河合 信明

【電話番号】 03-3454-1111

## 【選任した代理人】

【識別番号】 100111637

## 【弁理士】

【氏名又は名称】 谷澤 靖久

【電話番号】 03-3454-1111

【手数料の表示】

【予納台帳番号】 191928

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0213988

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 電子計算機、半導体集積回路、制御方法、プログラムの生成方法、及びプログラム

【特許請求の範囲】

【請求項 1】 論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置と処理装置の指定したコマンドを実行する制御装置とを有し、前記コマンドは、処理装置が所定の条件を検出した際に実行を指示され、再構成可能ハードウェアを論理形成するプログラムの切り換えを実行するためのコマンドを含むことを特徴とする電子計算機。

【請求項 2】 前記処理装置は、再構成可能ハードウェアを有する処理要素と前記再構成可能ハードウェアの論理回路を形成するプログラムを保持する 1 以上のプログラムデータメモリとを有する複数のバンクと、複数のバンクから 1 つを選択して有効とし外部と接続する有効バンク選択部とを有することを特徴とする請求項 1 の電子計算機。

【請求項 3】 前記処理装置は、再構成可能ハードウェアを含む処理要素と前記再構成可能ハードウェアの論理回路を形成するプログラムを保持する複数のプログラムデータメモリと複数のプログラムデータメモリから 1 つを選択して有効とする有効ブロック選択部とを含むバンクを有することを特徴とする請求項 1 の電子計算機。

【請求項 4】 前記処理装置は、少なくとも 1 つの処理要素は再構成可能ハードウェアで構成され、残りの処理要素は再構成可能ハードウェア又は汎用 CPU で構成されることを特徴とする請求項 2 又は 3 の電子計算機。

【請求項 5】 前記制御装置は、前記バンクが複数ある場合は有効とする前記バンクを指定するとともに前記プログラムデータメモリが複数ある場合は有効とする前記プログラムデータメモリを指定し指定した前記処理要素の動作を開始する `activate` コマンドと、指定した前記処理装置の動作を停止する `halt` コマンドと、前記制御装置から指定した前記処理装置へ割り込みベクタを発行する `interrupt` コマンドと、指定した記憶装置から前記プログラムデータメモリへプログラムデータを転送する `load__prg` コマンドと、`load`

d\_\_prg命令を中止するcancel\_\_prgコマンドと、load\_\_prg命令の終了まで待機するwait\_\_prgコマンドとを解釈実行することの特徴とする請求項2、3又は4の電子計算機。

【請求項6】 前記制御装置が実行するコマンドを保持するコマンドコードメモリを有し、前記制御装置は前記処理装置から指示されたアドレスに従ってコマンドコードメモリからコマンドを読み出して解釈実行するコマンドコード参照装置を有することを特徴とする請求項1、2、3、4又は5の電子計算機。

【請求項7】 前記コマンドコード参照装置は前記コマンドコードメモリのアドレスを保持するアドレスカウンタを有し、前記処理装置と制御装置間のコマンドの受渡しにおいて、前記処理装置が出力したアドレス信号線が有効であることを示す第1のアドレス制御線と、第1の制御線が有効の場合アドレス信号線の値をそのままアドレスカウンタに格納するか或いはアドレス信号線の値をアドレスカウンタの値に加算してアドレスカウンタに格納するかを指示する第2のアドレスカウンタ制御線を備えることを特徴とする請求項6の電子計算機。

【請求項8】 前記コマンドは、コマンドを分類するコマンドコードとアドレスカウンタ制御コードと後続のコマンドを実行するかどうかを示すフラグとからなるフォーマットで前記コマンドコードメモリに格納され、前記アドレスカウンタ制御コードは、アドレスカウンタに値を設定するload\_\_adrコマンドと、アドレスカウンタに指定した値を加算するadd\_\_adrコマンドとを含むこと特徴とする請求項7の電子計算機。

【請求項9】 前記アドレスカウンタ制御コードは、アドレスカウンタを前記制御装置に設けたアドレスカウンタスタックに退避するとともに新たな値をアドレスカウンタに設定するpush\_\_adrコマンドと、アドレスカウンタスタックの値をアドレスカウンタに戻すpop\_\_adrコマンドとを含むこと特徴とする請求項8の電子計算機。

【請求項10】 前記処理装置に対して転送するデータを一時的に保持するキャッシュメモリとキャッシュメモリを制御するキャッシュコントローラを含むキャッシュ装置を有し、前記処理装置が発行するコマンドによってキャッシュコントローラの制御を行なうことを特徴とする請求項1乃至9のいずれかの電子計

算機。

【請求項 11】 前記キャッシュ装置は前記処理装置の外部で定義するアドレスを処理装置の内部で定義されるアドレスに変換するアドレス変換装置を有し、前記処理装置が発行するコマンドによってアドレス変換装置の制御を行なうことを特徴とする請求項 10 の電子計算機。

【請求項 12】 論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置と処理装置の指定したコマンドを実行する制御装置とを有し、前記コマンドは、処理装置が所定の条件を検出した際に実行を指示され、再構成可能ハードウェアを論理形成するプログラムの切り換えを実行するためのコマンドを含み、

前記処理装置は、論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む第 2 の処理装置と第 2 の処理装置の指定したコマンドを実行する第 2 の制御装置から構成されることを特徴とする電子計算機。

【請求項 13】 請求項 1 乃至 11 のいずれかの電子計算機を実装した半導体集積回路。

【請求項 14】 論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置が所定の条件を検出した際にコマンドの実行を指示し、処理装置のコマンド実行指示を受けた制御装置が再構成可能ハードウェアを論理形成するプログラムの切り換えを実行することを特徴とする制御方法。

【請求項 15】 再構成可能ハードウェアと再構成可能ハードウェアの論理回路を形成するプログラムを保持する複数のプログラムデータメモリと複数のプログラムデータメモリから 1 つを選択して有効とする有効ブロック選択部とを含む制御装置が所定の条件を検出した際にコマンドの実行を指示し、処理装置のコマンド実行指示を受けた制御装置が、有効ブロック選択部を制御して指定されたプログラムデータメモリを有効にして再構成可能ハードウェアに接続する `activate` コマンドを実行し、再構成可能ハードウェアが実行する論理回路の内容を切り換えることを特徴とする制御方法。

【請求項 16】 前記制御装置は、指定した前記処理装置の動作を停止する `halt` コマンドと、前記制御装置から指定した前記処理装置へ割り込みベクタ

を発行する `interrupt` コマンドと、指定した記憶装置から前記プログラムデータメモリへプログラムデータを転送する `load__prg` コマンドと、`load__prg` 命令を中止する `cancel__prg` コマンドと、`load__prg` 命令の終了まで待機する `wait__prg` コマンドとを実行することを特徴とする請求項 15 の制御方法。

【請求項 17】 アプリケーションプログラムの制御フローを解析して、アプリケーションプログラムを処理単位に分割するとともに分割した処理単位を電子計算機内で実行する再構成可能ハードウェアが制御するコマンドを組み合わせたコマンドシーケンス中間コードを生成する制御フロー解析手順と、コマンドシーケンス中間コードを電子計算機で実行可能な形式に変換してコマンドシーケンスを生成するコマンドシーケンス実装手順と、処理単位の動作内容を電子計算機で実行可能な形式に変換するプログラムデータ生成手順とを有することを特徴とするプログラムの生成方法。

【請求項 18】 前記制御フロー解析手順は、アプリケーションプログラムの制御フローを解析して処理単位に分割する際に、各処理単位を前記再構成可能ハードウェアの論理を形成するプログラムを保持するプログラムデータメモリに格納可能なように分割することを特徴とする請求項 17 のプログラムの生成方法。

【請求項 19】 論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置が所定の条件を検出した際にコマンドの実行を指示すると、処理装置のコマンド実行指示を受けた制御装置が再構成可能ハードウェアを論理形成するプログラムの切り換えを実行する手順をコンピュータに実行させるプログラム。

【請求項 20】 再構成可能ハードウェアと再構成可能ハードウェアの論理回路を形成するプログラムを保持する複数のプログラムデータメモリと複数のプログラムデータメモリから 1 つを選択して有効とする有効ブロック選択部とを含む制御装置が所定の条件を検出した際にコマンドの実行を指示すると、処理装置のコマンド実行指示を受けた制御装置が、有効ブロック選択部を制御して指定されたプログラムデータメモリを有効にして再構成可能ハードウェアに接続を切り

換える `activate` コマンドを実行する手順をコンピュータに実行させるプログラム。

【請求項 21】 指定した前記処理装置の動作を停止する `halt` コマンド、前記制御装置から指定した前記処理装置へ割り込みベクタを発行する `interrupt` コマンド、指定した記憶装置から前記プログラムデータメモリへプログラムデータを転送する `load__prg` コマンド、`load__prg` 命令を中止する `cancel__prg` コマンド、或いは `load__prg` コマンドの終了まで待機する `wait__prg` コマンドを実行する手順をコンピュータに実行させる請求項 20 のプログラム。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

本発明は、電子計算機、半導体集積回路、制御方法、プログラムの生成方法、及びプログラムに関し、アプリケーションプログラムによる処理の一部分もしくはすべてを再構成可能なハードウェアを使用して高速に実行させるための技術に関する。

##### 【0002】

#### 【従来の技術】

従来、CPU の処理能力を越えるようなアプリケーションプログラムを実行するために、専用ハードウェアを用いる手法が提案／開発されている。図 30 はアプリケーションプログラムの全てを専用ハードウェア化する手法である。図 31 はアプリケーションプログラムの一部を専用ハードウェア化し、汎用 CPU とネットワークを介して接続して処理し、処理の一部を専用ハードウェアにより高速化する手法である。図 32 はアプリケーションプログラムの一部を専用ハードウェア化し、CPU の内部に専用ハードウェアで処理を実行する新たな命令セットを追加することで追加した命令で処理する部分を高速化する手法である。

このように、ハードウェアにアプリケーション全体もしくは一部を実装する手法は、処理能力を大幅に向上する反面、アプリケーション毎に新規にハードウェアを開発／製造する必要があるため、膨大な費用を必要としている。



一方、FPGA (field programmable gate array) や PLD (programmable logic device) に代表されるような論理回路をプログラムにより形成可能な再構成可能ハードウェアはプログラムを変更して論理回路を再構成することで、デバイスを取り替えることなくプログラムで指定した特定の処理を実行することができ、注目されている。

近年では、専用ハードウェアの代わりとしてこの再構成可能ハードウェアを用いることで、ハードウェアを新規に製造することなく、つまり低コストで、高い処理能力を必要とするアプリケーションプログラムを再構成可能ハードウェアで実現する方法及びその装置が提案されている。

例えば、特開平08-316329や特開平11-184718号公報では専用ハードウェアの代わりに再構成可能ハードウェアを用いている。また、特許3099889号公報ではCPUに付加する拡張命令を再構成可能ハードウェアで実現している。

#### 【0003】

##### 【特許文献1】

特開平08-316329号公報

##### 【特許文献2】

特開平11-184718号公報

##### 【特許文献3】

特許3099889号公報

##### 【特許文献4】

特開2001-147802号公報

##### 【特許文献5】

特表平11-507478号公報

#### 【0004】

##### 【発明が解決しようとする課題】

上記、従来の手法では、再構成可能ハードウェアに形成する論理回路の容量を考慮していないため、再構成可能ハードウェアの容量を越えるようなアプリケーションプログラムを実装することができない。従って、アプリケーションプログ

ラムの規模が大きくなればなるほどそれを実装するための再構成可能ハードウェアの規模も大きくなり高コストとなる。

#### 【0005】

一方、低コスト化のために、容量の小さい再構成可能ハードウェアにアプリケーションプログラムを実装する場合には、ハードウェアリソースに実装できる論理回路のサイズを意識してアプリケーションプログラムを分割する必要がある。しかしながら、分割したプログラムやそのプログラム間の制御は実装するアーキテクチャに強く依存するという問題を持つ。従って、分割したプログラムやそのプログラム間の制御はアーキテクチャが変わると再利用することができなくなり、プログラムの設計効率を著しく低下させる。

#### 【0006】

さらに、アプリケーションプログラムの分割の妥当性（再構成可能ハードウェアに実装可能かどうか）は再構成可能ハードウェアとして実現する機能の最終段階、すなわち再構成可能ハードウェアへの論理回路のマッピングの段階で判明するため、例えば実装可能なサイズを越えてしまうような、不具合のある分割は大幅な設計の後戻りとなり、設計効率が著しく低下する。

#### 【0007】

また、特開 2 0 0 1 - 1 4 7 8 0 2 号公報や特表平 1 1 - 5 0 7 4 7 8 号公報のように、分割した処理の間の制御を再構成可能ハードウェアの外部の CPU で制御する場合では、分割した処理間の制御は CPU のプログラムデータとして再利用可能であるが、分割した処理自体は、実装した再構成可能ハードウェアのアーキテクチャや実装可能な容量に依存してしまうため、再利用できない。さらに、処理間の制御を CPU で行なうため、CPU と再構成可能ハードウェアとの間でシステムコールのオーバーヘッドなどの待ち時間が生じてしまい、性能が低下する。これらの問題はアプリケーションプログラムの規模が大きくなるほど顕著に現れてくる。

#### 【0008】

本発明の目的は、アプリケーションプログラムを処理単位に分割して処理単位毎に切り換えて再構成可能ハードウェアに論理回路を構築して実行させることに

より低コストで処理速度を向上させ、アプリケーションプログラムを容易に再利用可能とした電子計算機、制御方法、プログラムの生成方法、及びプログラムを提供することにある。

#### 【0009】

##### 【課題を解決するための手段】

本発明の第1の電子計算機は、論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置と処理装置の指定したコマンドを実行する制御装置とを有し、前記コマンドは、処理装置が所定の条件を検出した際に実行を指示され、再構成可能ハードウェアを論理形成するプログラムの切り換えを実行するためのコマンドを含むことを特徴とする。

#### 【0010】

本発明の第2の電子計算機は、本発明の第1の電子計算機において、前記処理装置は、再構成可能ハードウェアを有する処理要素と前記再構成可能ハードウェアの論理回路を形成するプログラムを保持する1以上のプログラムデータメモリとを有する複数のバンクと、複数のバンクから1つを選択して有効とし外部と接続する有効バンク選択部とを有することを特徴とする。

#### 【0011】

本発明の第3の電子計算機は、本発明の第1の電子計算機において、前記処理装置は、再構成可能ハードウェアを含む処理要素と前記再構成可能ハードウェアの論理回路を形成するプログラムを保持する複数のプログラムデータメモリと複数のプログラムデータメモリから1つを選択して有効とする有効ブロック選択部とを含むバンクを有することを特徴とする。

#### 【0012】

本発明の第4の電子計算機は、本発明の第2又は第3の電子計算機において、前記処理装置は、少なくとも1つの処理要素は再構成可能ハードウェアで構成され、残りの処理要素は再構成可能ハードウェア又は汎用CPUで構成されることを特徴とする。

#### 【0013】

本発明の第5の電子計算機は、本発明の第2、第3、又は第4の電子計算機に

において、前記制御装置は、前記バンクが複数ある場合は有効とする前記バンクを指定するとともに前記プログラムデータメモリが複数ある場合は有効とする前記プログラムデータメモリを指定し指定した前記処理要素の動作を開始する `activate` コマンドと、指定した前記処理装置の動作を停止する `halt` コマンドと、前記制御装置から指定した前記処理装置へ割り込みベクタを発行する `interrupt` コマンドと、指定した記憶装置から前記プログラムデータメモリへプログラムデータを転送する `load__prg` コマンドと、`load__prg` 命令を中止する `cancel__prg` コマンドと、`load__prg` 命令の終了まで待機する `wait__prg` コマンドとを解釈実行することを特徴とする。

#### 【0014】

本発明の第6の電子計算機は、本発明の第1、第2、第3、第4、又は第5の電子計算機において、前記制御装置が実行するコマンドを保持するコマンドコードメモリを有し、前記制御装置は前記処理装置から指示されたアドレスに従ってコマンドコードメモリからコマンドを読み出して解釈実行するコマンドコード参照装置を有することを特徴とする。

#### 【0015】

本発明の第7の電子計算機は、本発明の第6の電子計算機において、前記コマンドコード参照装置は前記コマンドコードメモリのアドレスを保持するアドレスカウンタを有し、前記処理装置と制御装置間のコマンドの受渡しにおいて、前記処理装置が出力したアドレス信号線が有効であることを示す第1のアドレス制御線と、第1の制御線が有効の場合アドレス信号線の値をそのままアドレスカウンタに格納するか或いはアドレス信号線の値をアドレスカウンタの値に加算してアドレスカウンタに格納するかを指示する第2のアドレスカウンタ制御線を備えることを特徴とする。

#### 【0016】

本発明の第8の電子計算機は、本発明の第7の電子計算機において、前記コマンドはコマンドを分類するコマンドコードとアドレスカウンタ制御コードと後続のコマンドを実行するかどうかを示すフラグとからなるフォーマットで前記コマンドコードメモリに格納され、前記アドレスカウンタ制御コードは、アドレスカ

ウンタに値を設定する `load__adr` コマンドと、アドレスカウンタに指定した値を加算する `add__adr` コマンドとを含むこと特徴とする。

【0017】

本発明の第9の電子計算機は、本発明の第8の電子計算機において、前記アドレスカウンタ制御コードは、アドレスカウンタを前記制御装置に設けたアドレスカウンタスタックに退避するとともに新たな値をアドレスカウンタに設定する `push__adr` コマンドと、アドレスカウンタスタックの値をアドレスカウンタに戻す `pop__adr` コマンドとを含むこと特徴とする。

【0018】

本発明の第10の電子計算機は、本発明の第1乃至第9のいずれかの電子計算機において、前記処理装置に転送するデータを一時的に保持するキャッシュメモリとキャッシュメモリを制御するキャッシュコントローラを含むキャッシュ装置を有し、前記処理装置が発行するコマンドによってキャッシュコントローラの制御を行なうことを特徴とする。

【0019】

本発明の第11の電子計算機は、本発明の第10の電子計算機において、前記キャッシュ装置は前記処理装置の外部で定義するアドレスを処理装置の内部で定義されるアドレスに変換するアドレス変換装置を有し、前記処理装置が発行するコマンドによってアドレス変換装置の制御を行なうことを特徴とする。

【0020】

本発明の第12の電子計算機は、論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置と処理装置の指定したコマンドを実行する制御装置とを有し、前記コマンドは、処理装置が所定の条件を検出した際に実行を指示され、再構成可能ハードウェアを論理形成するプログラムの切り換えを実行するためのコマンドを含み、

前記処理装置は、論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む第2の処理装置と第2の処理装置の指定したコマンドを実行する第2の制御装置から構成されることを特徴とする。

【0021】

本発明の半導体集積回路は、本発明の第1乃至第11のいずれかの電子計算機を実装したことを特徴とする。

#### 【0022】

本発明の第1の制御方法は、論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置が所定の条件を検出した際にコマンドの実行を指示し、処理装置のコマンド実行指示を受けた制御装置が再構成可能ハードウェアを論理形成するプログラムの切り換えを実行することを特徴とする。

#### 【0023】

本発明の第2の制御方法は、再構成可能ハードウェアと再構成可能ハードウェアの論理回路を形成するプログラムを保持する複数のプログラムデータメモリと複数のプログラムデータメモリから1つを選択して有効とする有効ブロック選択部とを含む制御装置が所定の条件を検出した際にコマンドの実行を指示し、処理装置のコマンド実行指示を受けた制御装置が、有効ブロック選択部を制御して指定されたプログラムデータメモリを有効にして再構成可能ハードウェアに接続する `activate` コマンドを実行し、再構成可能ハードウェアが実行する論理回路の内容を切り換えることを特徴とする。

#### 【0024】

本発明の第3の制御方法は、本発明の第2の制御方法において、前記制御装置は、指定した前記処理装置の動作を停止する `halt` コマンドと、前記制御装置から指定した前記処理装置へ割り込みベクタを発行する `interrupt` コマンドと、指定した記憶装置から前記プログラムデータメモリへプログラムデータを転送する `load__prg` コマンドと、`load__prg` 命令を中止する `cancel__prg` コマンドと、`load__prg` 命令の終了まで待機する `wait__prg` コマンドとを実行することを特徴とする。

#### 【0025】

本発明の第1のプログラムの生成方法は、アプリケーションプログラムの制御フローを解析して、アプリケーションプログラムを処理単位に分割するとともに分割した処理単位を電子計算機内で実行する再構成可能ハードウェアが制御するコマンドを組み合わせたコマンドシーケンス中間コードを生成する制御フロー解

析手順と、コマンドシーケンス中間コードを電子計算機で実行可能な形式に変換してコマンドシーケンスを生成するコマンドシーケンス実装手順と、処理単位の動作内容を電子計算機で実行可能な形式に変換するプログラムデータ生成手順とを有することを特徴とする。

#### 【0026】

本発明の第2のプログラムの生成方法は、本発明の第1のプログラムの生成方法において、前記制御フロー解析手順は、アプリケーションプログラムの制御フローを解析して処理単位に分割する際に、各処理単位を前記再構成可能ハードウェアの論理を形成するプログラムを保持するプログラムデータメモリに格納可能なように分割することを特徴とする。

#### 【0027】

本発明の第1のプログラムは、論理回路をプログラムにより形成可能な再構成可能ハードウェアを含む処理装置が所定の条件を検出した際にコマンドの実行を指示すると、処理装置のコマンド実行指示を受けた制御装置が再構成可能ハードウェアを論理形成するプログラムの切り換えを実行する手順をコンピュータに実行させる。

#### 【0028】

本発明の第2のプログラムは、再構成可能ハードウェアと再構成可能ハードウェアの論理回路を形成するプログラムを保持する複数のプログラムデータメモリと複数のプログラムデータメモリから1つを選択して有効とする有効ブロック選択部とを含む制御装置が所定の条件を検出した際にコマンドの実行を指示すると、処理装置のコマンド実行指示を受けた制御装置が、有効ブロック選択部を制御して指定されたプログラムデータメモリを有効にして再構成可能ハードウェアに接続を切り換える `activate` コマンドを実行する手順をコンピュータに実行させる。

#### 【0029】

本発明の第3のプログラムは、本発明の第2のプログラムにおいて、指定した前記処理装置の動作を停止する `halt` コマンド、前記制御装置から指定した前記処理装置へ割り込みベクタを発行する `interrupt` コマンド、指定した

記憶装置から前記プログラムデータメモリへプログラムデータを転送する `load__prg` コマンド、`load__prg` 命令を中止する `cancel__prg` コマンド、或いは `load__prg` コマンドの終了まで待機する `wait__prg` コマンドを実行する手順をコンピュータに実行させる。

### 【0030】

#### 【発明の実施の形態】

本発明の実施の形態について図面を参照して詳細に説明する。図1は本発明の実施の形態の電子計算機の基本的な構成を示すブロック図である。本発明の第1の実施形態の電子計算機30は、外部とのインタフェースを制御しデータを転送するインタフェース装置40と、アプリケーションプログラムの全部又は一部の処理を実行する処理装置70と、処理装置70から指示されたコマンドを実行する制御装置60とを含み、接続網20を介して外部記憶装置10と接続する。図1の電子計算機30は、処理装置70を2つ含んでいるが、1つでもよく、3つ以上でもよい。

### 【0031】

処理装置70は、メモリ部80と処理部90からなる。図2は処理装置70の構成の一例を示した図であり、1つの処理要素91と1つのプログラムデータメモリ81からなるバンク101と、1つの処理要素91に対して複数のプログラムデータメモリ81を設けて選択できるようになっているバンク102と、選択メモリ83と、複数の処理要素91から1つを選択して出力する有効ブロック選択部82を構成要素とする。

### 【0032】

図2では処理装置70はバンク101とバンク102をそれぞれ2つずつ含んでいるが、バンク101又はバンク102のいずれかを1つ以上含めばよい。また、バンクが1つの構成であれば有効ブロック選択部82は省略でき、バンクもプログラムデータメモリ81も1つの構成であれば選択メモリ83を省略できる。

### 【0033】

処理装置70に入力されるプログラムデータ信号S80は、プログラムデータ



メモリ 81 及び選択メモリ 83 に接続される。プログラムデータメモリ 81 は処理要素 91 の処理内容を決定するためのプログラムを保持するメモリである。プログラムデータメモリ 81 に保持されるプログラムは通常外部記憶装置 10 に格納されており、必要に応じて接続網 20、インタフェース装置 40、プログラムデータ信号 S80 を介して転送されプログラムデータメモリ 81 に書き込まれる。なお、処理要素 91 が F P G A のような再構成可能ハードウェアである場合、プログラムデータメモリ 81 へ保持するプログラムは処理要素 91 の論理回路の形成を行うためのプログラムとなる。

#### 【0034】

処理要素 91 は接続するプログラムデータメモリ 81 に保持されるプログラムの内容に従った処理を実行する。例えば図 3 のように再構成可能ハードウェアで実現されるが、図 4 に示すように C P U にて実現してもよい。

#### 【0035】

選択メモリ 83 は、有効バンク選択部 92 及び有効ブロック選択部 82 に接続され、処理装置 70 の中で有効にするバンクを選択したり、バンク 102 のように複数のプログラムデータメモリ 81 で構成されたバンクの中で有効にするプログラムデータメモリ 81 を選択したりする情報を保持する。なお、プログラムデータメモリ 81 への切り換えは瞬時に終わるので、切り換えられたプログラムデータメモリ 81 へのプログラムの格納が終了していれば瞬時に新たなプログラムに対応した処理を処理要素 91 は開始することができる。ただし、プログラムの格納が終わっていなければ、データ転送が終了しプログラムが格納されるまで待つ必要がある。

#### 【0036】

処理装置 70 から出力されるコマンド信号 S91 は処理装置 70 の処理要素 91 から生成される。処理装置 70 に入力される割り込み信号 S92 は処理装置 70 の処理要素 91 に入力され、その処理の過程で使用される。処理装置 70 に出力される処理データ信号 S93 は、処理部 90 に接続され、処理に必要なデータや、処理を終えたデータの入出力線として使用される。

#### 【0037】

コマンド信号 S 9 1、割り込み信号 S 9 2 及び処理データ信号 S 9 3 は処理装置 7 0 が複数のバンクを含む場合選択メモリ 8 3 によって有効になっているバンクの処理装置 9 0 に入出力される。図 2 では、処理装置 7 0 へ入力したデータはすべての処理装置 9 0 に接続しているが、有効ではないバンクに入力しないように、選択メモリ 8 3 等から制御してそれらの入力データを切断するようにしてもよい。

#### 【0038】

次に、各構成要素 9 1 について詳細に説明する。処理要素 9 1 は図 3 に示すように、従来の F P G A や P L D に代表されるような再構成可能ハードウェアで構成される。ただし、構成要素 9 1 が複数ある場合に、その中の 1 つ又は複数を図 4 に示すように、C P U 1 2 0 で構成してもよい。C P U 1 2 0 で構成した構成要素 9 1 はアプリケーションプログラムの処理を高級言語で処理する部分を割り当てて処理したり、処理装置 7 0 内を制御したりすることを主な用途とする。なお、処理要素 9 1 を C P U 1 2 0 で構成した場合には、処理データ信号 S 9 3 を介してプログラムを読み込めるため、図 4 におけるプログラムデータ接続線 S 1 0 1 はなくてもよい。

#### 【0039】

図 5 に示すように、本電子計算機 3 0 における制御装置 6 0 を再構成可能なハードウェアで実現し、再構成可能制御装置 R 6 0 を構成してもよい。図 5 に示したように、再構成可能制御装置 R 6 0 はプログラムデータ接続線 S 1 0 1 を用いて設定され、設定される内容に従って処理内容を変えることができる。またコマンド信号 S 9 1 や割り込み信号 S 9 2 及び処理データ信号 S 9 3 はインタフェース装置 4 0 を介して入出力される。

#### 【0040】

インタフェース装置 4 0 は処理装置 7 0 や制御装置 6 0 と電子計算機 3 0 の外部の接続網 2 0 とを接続し、接続網 2 0 側から電子計算機 3 0 に制御に関する通信が発生した場合には、接続網 2 0 の適切なプロトコルに基づいて制御装置 6 0 にコマンド信号 S 4 1 を出力する。制御装置 6 0 からインタフェース装置 4 0 に割り込み信号 S 4 2 等の制御に関する通信が発生した場合には、同様に適切なプ

ロトコルを用いて接続網 20 を介して指定された接続先に伝える。

#### 【0041】

電子計算機 30 の内部から外部へのアクセスが発生した場合は、インタフェース装置 40 が適切なプロトコルに基づいて外部にアクセスを行なう。電子計算機 30 の外部から処理装置 70 に対するアクセスが発生した場合には、インタフェース装置 40 が適切なプロトコルに基づいてアクセスする。制御装置 60 は、インタフェース装置 40 を介して電子計算機 30 の外部の装置から伝えられたコマンド信号 S41 や処理装置 70 から出力されたコマンド信号 S91 を受けとり、受け取ったコマンドを解釈実行する。処理装置 70 が制御装置 60 へコマンド信号 S91 を発行する際のプロトコルの例を図 6 に示す。このプロトコルはインタフェース装置 40 と制御装置 60 の間のプロトコルに適用してもよい。

#### 【0042】

図 6 は、処理装置 70 がリクエスト信号 S911 と共に直接コマンドコード信号 S912 を渡す方式を示す図である。コマンドコード信号 S912 を受けとった制御装置 60 はコマンドコードの内容に従って処理を行ない、処理終了時に応答信号 S921 を返す。

#### 【0043】

制御装置 60 が解釈実行するコマンドコードの一例を図 7 及び図 8 に示す。図 7 はコマンドコードの構成を示した図であり、コマンドコード A10 はコマンドコード名 A11 とコマンドコードパラメータ A12 で構成される。図 8 はコマンドの実行内容を一覧にした図であり以下に説明する 6 つのコマンドを示している。

#### 【0044】

activate は、選択メモリ 83 に、コマンドコードパラメータ A12 で指示されるコードを書き込むことで、有効バンク選択部 92 及び有効ブロック選択部 82 を制御し、選択したプログラムデータメモリ 81 を同じバンクの処理要素 91 に接続する。例えば図 3 に示したように、再構成可能ハードウェアで処理要素 91 が構成されている場合に activate することは、再構成可能ハードウェアにそのプログラムデータを設定することを意味し、activate さ

れた再構成ハードウェアはすぐにそのプログラムデータメモリ 81 の内容に従って処理を開始する。

#### 【0045】

`halt` は、コマンドコードパラメータ A12 によって指定した処理装置 70 の動作を停止させる。

`interrupt` は、コマンドコードパラメータ A12 によって指定した処理装置 70 に対して、指定した割り込みベクタ信号 S922 を発行する。

`load_prg` は、コマンドコードパラメータ A12 によって指定したプログラムデータメモリ 81 の領域へ、外部記憶装置 10 やその他任意の記憶装置に格納されているプログラムデータを転送する。

`cancel_prg` は、`load_prg` によって開始した転送を中止する。

`wait_prg` は、`load_prg` によって開始した転送が終了するまで待機する。

#### 【0046】

制御装置 60 は、図 8 に示したコマンドの任意の組合せ（コマンドセット）を解釈し、それぞれのコマンドに対して適切な処理を行ない、応答信号を含む割り込みを処理装置 70 や、インタフェース装置 40 を介して電子計算機 30 の外部に出力する。コマンドの解析や処理、割り込みは、処理装置 70 毎に並列に行なってもよい。

#### 【0047】

処理装置 70 が制御装置 60 へコマンド信号 S91 を発行する際の別のプロトコルの例を図 9 及び図 11 に示す。このプロトコルはインタフェース装置 40 と制御装置 60 の間のプロトコルに適用してもよい。

#### 【0048】

図 9 は、制御装置 60 がコマンドコード参照装置 61 とコマンドコードメモリ 63 を有し、処理装置 70 がリクエスト信号 S911 と共にアドレス信号 S913 を渡す方式を示す図である。図 10 はコマンドコードメモリ 63 に格納されるコマンドコードの一例を示す図である。アドレス信号 S913 は、制御装置 60 に実行させたいコマンドが格納されたコマンドコードメモリ 63 のアドレスを指

す。

#### 【0049】

アドレス信号 S913 を受けとった制御装置 60 は、コマンドコード参照装置 61 を用いて、コマンドコードメモリ 63 を参照し、アドレス信号 S913 に対応するコマンドを実行後、処理終了時に応答信号 S921 を返す。

#### 【0050】

コマンドコードメモリ 63 は、制御装置 60 から参照可能な、制御装置 60 外部のメモリや外部記憶装置 10 等の任意のメモリでもよい。また、処理装置 70 が複数ある場合は制御装置 60 内にコマンドコード参照装置 61 を複数持つてコマンドを並列処理するようにしてもよいし、処理装置 70 に対応してコマンドコード参照装置 61 を設けるように構成してもよい。

#### 【0051】

図 9 のプロトコルは、制御装置 60 にコマンドコード参照装置 61 とコマンドコードメモリ 63 を設ける必要があるが、一般に、メモリのアドレスのビット数はデータのビット数より少なくすることができるため、図 6 のプロトコルに比べて、制御装置 60 と処理装置 70 との接続信号線数を少なくすることができる。

#### 【0052】

図 11 は、制御装置 60 がコマンドコード参照装置 61 及びそのアドレスカウンタ 62 を有し、処理装置 70 がリクエスト信号 S911 と共にアドレス操作信号 S914 及び S915 を用いてアドレスカウンタ 62 の制御を行ない、必要ならばアドレス信号 S913 を用いてアドレスを制御装置 60 に渡す方式を示した図である。コマンドコード参照装置 61 及びそのアドレスカウンタ 62 は接続する処理装置 70 毎に設けてもよい。

#### 【0053】

`adr__ena` アドレスカウンタ操作信号 S914 が有効でかつ `direct` / `offset` アドレスカウンタ操作信号 S915 が `direct` を示す場合には、処理装置 70 から渡されるアドレス信号 S913 をアドレスカウンタ 62 に格納する。`adr__ena` アドレスカウンタ操作信号 S914 が有効でかつアドレスカウンタ操作信号 S915 が `offset` を示す場合には、処理装置 70 か

ら渡されるアドレス信号 S 9 1 3 の値をアドレスカウンタ 6 2 に加算する。a d r \_ e n a アドレスカウンタ操作信号 S 9 1 4 が無効の場合には、処理装置 7 0 から渡されるアドレス信号 S 9 1 3 は無視し、アドレスカウンタ 6 2 の値は保持する。

#### 【0054】

処理装置 7 0 によるアドレスカウンタ 6 2 の制御が終ると、制御装置 6 0 はアドレスカウンタ 6 2 の値とコマンドコード参照装置 6 1 を用いてコマンドが格納されたコマンドコードメモリ 6 3 を参照して格納されているコマンドの処理を行ない、終了後に応答信号 S 9 2 1 を返す。

#### 【0055】

図 1 2 は図 1 1 の構成におけるコマンドコードメモリ 6 3 に格納されるコマンドコードの一例を示した図である。図 1 2 に示すように、コマンドコードメモリ 6 3 には、コマンドコード A 1 0 以外にアドレスカウンタ制御コード A 2 0 及びフラグ A 3 0 を格納しているが、含まないようにしてもよい。図 1 2 のようなフォーマットを用いた場合、制御装置 6 0 は、コマンドコード A 1 0 に記された処理の終了後、処理装置 7 0 に応答信号 S 9 2 1 を返す前にアドレスカウンタ制御コード A 2 0 に指定されている処理を行なう。

#### 【0056】

図 1 3 にアドレスカウンタ制御コード A 2 0 の詳細を示す。アドレスカウンタ制御コード A 2 0 はアドレスカウンタ制御コード名 A 2 1、そのパラメータであるアドレスカウンタ制御コードパラメータ A 2 2 から構成される。図 1 4 にアドレスカウンタ制御コードの例を示す。l o a d \_ a d r はアドレスカウンタ制御コードパラメータ A 2 2 の値を新しいアドレスカウンタ 6 2 の値として設定する。a d d \_ a d r はアドレスカウンタ制御コードパラメータ A 2 2 の値をアドレスカウンタ 6 2 の値に加算する。p u s h \_ a d r は現在のアドレスカウンタ 6 2 の値を図示しないがアドレスカウンタスタックに保存し、アドレスカウンタ制御コードパラメータ A 2 2 の値を新しいアドレスカウンタ 6 2 の値として設定する。p o p \_ a d r 命令はアドレスカウンタスタックから値を取り出して、その値を新しいアドレスカウンタ 6 2 の値として設定する。アドレスカウンタスタック

クはコマンドコード参照装置 61 に設ければよい。

#### 【0057】

また、フラグ A30 は、アドレスカウンタ制御コード A20 を実行後の新しいアドレスカウンタ 62 の値を用いて、引続きコマンドの参照及び実行を行なうかどうかのフラグとして用いる。以後、引続きコマンドを実行する場合のフラグを `cont`、実行しない場合のフラグを `stop` と記す。

例えば、図 11 に示した構成において、コマンドコードメモリ 63 が図 15 のようなコマンドシーケンスを格納し、処理装置 70 がアドレス信号 S913 を 100 に、アドレスカウンタ操作信号 S914 を無効にしてコマンドを発行すると制御装置 60 はコマンドコードを Y100、Y101、Y200 の順に実行する。

#### 【0058】

従って、コマンドコード参照装置 61 がアドレスカウンタ 62 を内蔵し、アドレスカウンタ制御コード A20 でアドレスカウンタ 62 の制御を行なう方式では、処理装置 70 はコマンド発行の際に必要な場合にだけアドレス信号 S913 を出力するだけで済むため、処理装置 70 内の処理部 90 は少ないハードウェアリソースでアドレス信号 S913 を生成できるようになる。

#### 【0059】

なお、処理装置 70 と制御装置 60 間のプロトコルに関しては、図 6、図 9、図 14 の中から 1 つを選択するか、或いはこれらを任意に組み合わせて構成してもよい。また、複数の構成を組み込んで必要に応じて制御線を追加してプロトコルを切替えるようにしてもよい。例えば、制御線を追加し、図 6 のプロトコルか図 11 のプロトコルかを選択できるようにしてもよい。

#### 【0060】

次に、電子計算機 30 と外部とのデータ転送を高速化するため、図 16 に示すように、図 1 の電子計算機 30 にキャッシュコントローラ 130 を含むキャッシュ装置 50 を追加して、そのキャッシュコントローラ 130 の制御をコマンドで行なうように構成してもよい。図 16 ではキャッシュ装置 50 はキャッシュコントローラ 130 を 3 つ設けそれぞれ制御装置 60、メモリ部 80、処理部 90 に接続しているが、1 つのキャッシュコントローラ 130 でこれらと接続する構成

でもよい。また、キャッシュコントローラ 130 内にキャッシュメモリ 140 を複数ポート持ってもよい。また、複数のキャッシュコントローラ間でアドレス変換装置 150 を共有してもよい。

#### 【0061】

キャッシュコントローラ 130 は、例えば外部記憶装置 10 等に記憶され処理装置 70 がアクセスするデータを一時的に保持するキャッシュメモリ 140 とアドレス変換装置 150 を含んでいる。キャッシュコントローラ 130 は、コマンドによって制御され、主にキャッシュメモリ 140 と外部記憶装置 10 との間のデータ転送と、キャッシュメモリ 140 と制御装置 60 や処理装置 70 との間のデータ転送を行ない、処理装置 70 や制御装置 60 と並列に動作する。

#### 【0062】

アドレス変換装置 150 は、処理装置 70 のアドレス空間とインタフェース装置 40 のアドレス空間との間のアドレスを互いに変換する装置であり、処理装置 70 内で独立したアドレス空間を持つことができる。また、処理装置 70 毎にアドレス変換装置 150 を設けることにより処理装置 70 毎に独立したアドレス空間を定義することもできる。

#### 【0063】

また、インタフェース装置 40 に設けられたアドレス変換装置 151 は、電子計算機 30 と、電子計算機 30 と接続網 20 を介して接続する外部記憶装置 10 や他の装置との間のアドレス空間の差分に対してアドレス変換する。処理装置 70 は制御装置 60 でコマンドを実行してアドレス変換装置 150 の制御を行なう。

#### 【0064】

なお、キャッシュコントローラを制御するためのコマンドについては図 8 に記載していないが、例えばアドレス変換装置 150 が被変換アドレスと変換アドレスとを対にして記憶する変換用バッファを備える場合はバッファへの登録、消去、置換等の制御を行うコマンド等を設ければよく、また、キャッシュメモリ 140 内の特定の領域を処理装置 70 の専用のローカルメモリ領域として設定するようなコマンドを追加してもよい。



**【0065】**

また、データフローを解析してスケジューリングを行ない、使用するキャッシュやそのキャッシュの制御を予めコマンドシーケンスとして記述しておくことで、処理装置70による処理と、制御装置60によるデータフローの制御を並列に行なうことができ、処理装置70の処理能力が向上する。また、処理装置70が必要なタイミングでコマンドを発行するため、無駄なオーバヘッドも発生しない。

**【0066】**

例えば、キャッシュコントローラ130をコマンドで制御して、各装置がキャッシュメモリ140にアクセスするより前に、前もってデータをキャッシュメモリ140にロードしておくことが可能になり、効率の良いデータ転送が可能になる。

なお、本電子計算機30を接続網20に多数接続しても互いに通信および制御が可能であることは、上記説明から明白であり、本電子計算機30は任意に拡張が可能である。また、本電子計算機30は、本電子計算機30を1つ以上含むLSIや、本電子計算機30の一部分をLSIとして実現してもよい。さらに、本電子計算機30を論理的にFPGAやPLD等の再構成可能ハードウェア上に論理的に実装してもよい。

**【0067】**

次に、本発明の実施形態の電子計算機30の動作を図面を参照して説明する。なお、電子計算機30の構成については様々な構成について説明してきたが、動作の説明では図17に示す構成を例として説明する。動作説明は電子計算機で処理する対象となるアプリケーションプログラムをプログラムデータメモリ811～813へ格納する処理単位へと分割し制御装置60で実行するコマンドを生成するプログラムの生成方法を説明し、その次に図17の構成における電子計算機30での動作・制御方法について説明する。

**【0068】**

図17を参照すると、3つのプログラムデータメモリ811, 812, 813に、有効ブロック選択82を介して接続され再構成可能ハードウェアで実現され

た処理要素 110 を 1 バンク有する 1 つの処理装置 71 があり、図 11 から direct/offset を指定するアドレスカウンタ制御 S915 を省略したプロトコルで制御装置 60 と処理装置 71 を接続し、制御装置 60 とインタフェース装置 40 との間のコマンド信号 S41 及び割り込み信号 S42 を省略した構成となっている。図 17 の制御装置 60 に実装されているコマンドセットは図 8 の通りとし、同制御装置 60 に実装されているアドレスカウンタ制御コードは図 13 及び図 14 の通りとする。なお、キャッシュ装置 50 は省略した構成としている。

#### 【0069】

図 18 は図 17 の処理要素 110 で実行するアプリケーションの処理内容の流れの一例を示した制御フロー図である。図 18 に示しているように、アプリケーションは初期状態 C0 を経て、制御フローで示している処理を動的に切り換えながら実行し、全ての処理が終ると終了状態 C9 となる。初期状態 C0 は処理 P1 を開始できる直前の状態とする。終了状態 C9 は、一例として図 17 における処理装置 71 がすべて停止した状態とする。

#### 【0070】

図 18 に示したアプリケーションの処理内容は、5 つの状態 (C1, C2, C3, C4, C5) と 4 種類の処理内容 (P1, P2, P3, P4) を持つものとし初期状態 C0 から状態 C1 には無条件で遷移する。状態 C1 では処理 P1 を行ない、条件 F1 で状態 2 に遷移する。状態 C2 では処理 P2 を行ない、条件 F2a で状態 3 に、条件 F2b で状態 4 に遷移する。状態 C3 では処理 P1 を行ない、条件 F3 で状態 5 に遷移する。状態 C4 では処理 P3 を行ない、条件 F4 で状態 5 に遷移する。状態 C5 では処理 P4 を行ない、条件 F5 で終了状態 C9 に遷移する。

#### 【0071】

図 19 はアプリケーションを実行するプログラムを生成するフローである。図 19 に示すフローは、各処理後に実行するコマンドシーケンスを生成する制御フロー解析手順 M1、コマンドシーケンスをデータ列に変換するコマンドシーケンス実装手順 M2、プログラムデータを生成するためのプログラムデータ生成手順

M3から成り、アプリケーション全体の制御フロー、終了、電子計算機30の構成情報及びそのコマンドセット等を入力とし、本電子計算機の初期状態C0を示すコマンドシーケンスコード、各処理装置内で使用する全てのプログラムデータ、及びそれらから参照されるコマンドシーケンスコードを出力する。制御フロー解析手順M1、コマンドシーケンス実装手順M2、プログラムデータ生成手順M3はそれぞれプログラムで実現される。

#### 【0072】

制御フロー解析手順M1は、各処理(P1～P4)とそれに対応する状態(C1～C5)、各状態に対応した遷移条件(F1～F5)と遷移先を解析する。そして、各処理をプログラムデータメモリ811～プログラムデータメモリ813のいずれかに割り当て、有効ブロック選択部82を切り換えて各処理を続けて実行することで次状態に遷移できるようなコマンドシーケンスの中間コードを生成する。

#### 【0073】

図18のアプリケーションを制御フロー解析手順M1によって解析した結果として生成されるコマンドシーケンスの中間コードの一例を図20に示す。ここで、電源投入時には、有効ブロック選択部82はプログラムデータメモリ811を選択しているものとする。図20において、コマンドシーケンスSQ0のコマンド”load\_\_prg 812、PM1”のパラメータである812は処理要素110において処理P1を実行するように生成されたプログラムデータのロード先がプログラムメモリデータメモリ812であることを示し、PM1は同プログラムデータが格納されているメモリの領域を示す。PM1で指定されるプログラムデータPM1は外部記憶装置10を含む任意のメモリに格納されていてよい。現段階では、プログラムデータPM1自体は生成されていないので空のメモリ領域を指している。プログラムデータPM2、PM3、PM4もそれぞれ処理P2、処理P3、処理P4を実行するように生成されたプログラムデータが格納されているメモリの領域を指定するものである。

#### 【0074】

図20において、例えば、状態C4は処理P3を実行している状態であり、こ

の状態で条件 F 4 が成立した時点でコマンドシーケンス S Q 4 を開始する。コマンドシーケンス S Q 4 の処理内容は、“wait\_\_prg 813、PM4”によりプログラムデータメモリ 813 にプログラムデータ PM4 が全て転送されるまで待機し、“activate”によりプログラムデータメモリ 813 を選択して開始することとなる。プログラムデータメモリ 813 が“activate”されると、処理要素 110 はプログラムデータメモリ 813 に保持されているプログラムで決められた処理を開始する。この時点では、プログラムデータメモリ 813 には、処理 PM4 を実行するプログラムデータが保持されているため、処理要素 110 は処理 PM4 を開始する。これは、すなわち、状態 C 5 に移行することを意味する。

#### 【0075】

このように、処理内容にコマンドシーケンスの実行手順を追加することで次の状態に遷移することができるようになる。しかも、処理装置（処理要素）自身が決められた条件を検出したタイミングでコマンドを出すことができるため、遷移のタイミングを処理内容に含めることができ、処理効率を向上させる。また、図 20 において、処理 P 1 を行なっている際には、状態が C 1 もしくは C 3 であるかを識別する必要があるため、処理 P 1 を行なうプログラムデータメモリを“activate”する前に、処理 P 1 を行なう処理装置 71 に割り込みベクタを設定し、その値を用いた遷移条件に修正している。例えば、状態 C 1 の割り込みベクタの設定は S Q 0 の“interrupt 71、C1”にて実行する。

#### 【0076】

コマンドシーケンス実装手順 M 2 を用いて、図 20 のコマンドシーケンスをメモリに割り当てた様子を図 22 に、各処理におけるコマンドシーケンスの呼び出しを図 17 の構成における制御装置 60 と処理装置 71 との間のインタフェースのプロトコルに変換した、各処理毎の制御フローを図 21 にそれぞれ示す。

#### 【0077】

図 21 及び図 22 では、電源投入後最初に実行するコマンドシーケンス S Q 0 A も含めて示している。コマンドシーケンス S Q 0 A を実行することにより初期状態 C 0 へ遷移する。状態 C 0 では有効ブロック選択部 82 はプログラムデータ

メモリ 811 を選択しているため、処理要素 110 はプログラムデータメモリ 811 に格納されているプログラムの動作を電源投入後に開始する。

#### 【0078】

図 21 に示したように、個々の処理は電源投入時及びオリジナルの処理 (P1, P2, P3, P4) に各コマンドシーケンス発行処理を追加した制御フローとなる。例えば、図 21 の処理 P2 において条件 F2a が成立した際にはコマンド発行処理 SQ2bA を実行し、条件 F2b が成立した際にはコマンド発行処理 SQ2bA を実行する。処理装置 71 がコマンド発行処理 SQ2aA を実行する際には、リクエスト信号 S911 と、adr\_\_ena アドレスカウンタ操作信号 S914 及び ADR002 を示すアドレス信号 S913 が出力される。図 22 より、アドレス ADR002 はコマンドシーケンス SQ2a を指すため、制御装置 60 は SQ2a の内容を実行する。

#### 【0079】

また、図 20 から分かるように、コマンドシーケンス SQ3 とコマンドシーケンス SQ4 は全く同じ動作をするため、図 22 に示したように、ベースアドレス値を ADR004 として 1 つに集約することができる。また、図 18 から分かるように、状態 C3 もしくは状態 C4 の後には状態 C5 となるため、図 22 では、コマンドシーケンス SQ3 及び SQ4 を指すアドレス ADR004 のオフセット +1 のアドレスカウンタ制御コードにおいて、"load\_\_adr" を使わずに "add\_\_adr" を使用して次のアドレスカウンタ 62 の値 (ADR004 オフセット +2) を設定している。このように設計することで、処理装置 71 は余分なアドレスを出力する必要がなくなる。

#### 【0080】

最後に、図 21 で示した個々の処理を、プログラムデータ生成手順 M3 を用いて、それぞれの処理を実行するプログラムデータを生成する。生成した各プログラムデータをメモリに格納した例を図 23 に示す。生成したプログラムデータ PM1、PM2、PM3、PM4 が格納されたメモリ領域は、図 22 の各コマンドシーケンスのパラメータに反映させる。

#### 【0081】

なお、図16に示したように、電子計算機にアドレス変換装置を備えることで、個々の処理でアクセスするメモリアドレスやコマンドが格納されたメモリを指すアドレスは独立したアドレス空間で設計することもできる。また、プログラムデータPM0に関しては、図17の構成例の電子計算機の電源投入直後に実行する必要があるため、このプログラムデータPM0は、あらかじめプログラムデータメモリ811に格納されている必要がある。従って、本実装例における、電源投入時の電子計算機の初期状態は図24のリストのようになる。

#### 【0082】

次に電子計算機30の動作・制御方法について説明する。図25は上記説明した動作をタイミングチャートとしてまとめて示した図25である。図25と図17、図18、図21、図22、図23、図24を用いて説明する。

#### 【0083】

図25は横軸に、制御装置60内のアドレスカウンタ62の値、制御装置60の動作内容、処理要素110の動作内容、処理要素110に入力される割り込みベクタ信号S922の内容、各プログラムデータメモリ811、812、813が保持しているプログラムデータの内容、プログラムデータの転送状態を示し、縦軸は下向きに経過時間をT101から順に表している。なお、プログラムデータメモリ811、812、813の部分を編み目で表示している部分は実行中であることを示している。

#### 【0084】

T101のタイミングは図24に示した電源投入時の電子計算機30の状態であり、プログラムデータメモリ811にコマンド発行処理SQ0Aを行なうプログラムデータPM0が格納されている。

#### 【0085】

T102で処理要素110はPM0の動作を開始し、コマンド発行処理SQ0Aを実行する。制御装置60のコマンド参照装置61は、コマンド発行処理SQ0Aを受けると、アドレスカウンタ62にアドレス値ADR000を設定し、図22のコマンドシーケンスが格納されたコマンドコードメモリ63のADR000に格納されたコマンドを読み出して、T103で"load\_\_prg"を実行

する。T103で制御装置60は、"load\_\_prg"を実行し、プログラムデータメモリ812へプログラムデータPM1を転送し始める。プログラムデータPM1の格納場所の情報は"load\_\_prg"のパラメータのPM1に含まれるものとし、ここでは外部記憶装置10に図23のように格納されているものとするが、外部記憶装置10以外のメモリに格納しておいてもよい。

#### 【0086】

図22を参照すると、ADR000のアドレスカウンタ制御コードは次のコマンドも実行するように設定されているため、制御装置60はアドレスカウンタ62に1を加え、T104で次の命令を実行する。T104で制御装置60は、プログラムデータメモリ812にプログラムデータPM1がすべて転送されるのを待ち、T105でその転送が終了すると、制御装置60は引続きT106で次の命令を実行する。

#### 【0087】

以下、同様に実行され、T017で"activate 812"を制御装置60が実行すると、選択メモリ83がプログラムデータメモリ812を選択する情報に更新され、有効ブロック選択82は選択メモリ83の指示によりプログラムデータメモリ812に切り換え、処理要素110は動作が切替えられてプログラムデータメモリ812に格納されているPM1の処理を開始する。T108で、制御装置60はアドレスカウンタ制御コード"add\_\_adr 0/stop"によりT102から開始した一連のコマンドシーケンスの処理を終了する。

#### 【0088】

以下、同様に実行され、T110において、処理要素110が処理P1を実行中に条件F1を検出し、割り込みベクタ信号S922がC1であることを確認するとコマンド発行処理SQ1Aを実行する。

T110でコマンド発行処理SQ1Aを受けた制御装置60はアドレスカウンタ62の値をADR001の値に更新する。

#### 【0089】

T112で、制御装置60はコマンドコードメモリ63を読み出して"activate 813"を実行して、処理要素110の処理内容をプログラムデー

タメモリ 813 に格納される PM2 の処理に切り換える。

#### 【0090】

以下同様に実行され、T123 において、処理要素 110 が処理 P4 を実行中に条件 F5 を検出するとコマンド発行処理 SQ5A を実行する。T123 でコマンド発行処理 SQ5A を受けた制御装置 60 はコマンドコードメモリ 63 を参照し、T124 で処理装置 71 対する "halt" を実行する。"halt" による停止指示を受けると処理装置 71 は動作を終了し、"halt" を実行した制御装置 60 もコマンドシーケンスの解釈実行を終了する。

#### 【0091】

このように、生成されたプログラムデータは PM0、PM1、PM2、PM3、PM4 の 5 つであるが、使用するプログラムデータメモリは 811、812、813 の 3 つである。これは、ハードウェアリソースを越えるようなアプリケーションが本電子計算機 30 に実装できることを示している。

#### 【0092】

また、図 17 の構成では処理装置 71 はバンクを 1 つしか含まない構成となっているが、2 つ以上のバンクを有する場合はプログラムデータ PM0 ~ PM4 を複数のバンクのプログラムデータメモリに格納しそれぞれの再構成可能ハードウェアで構成された処理要素で実行させるようにしてもよい。この場合、処理装置の出力は有効バンク選択部 92 で選択された処理要素のコマンドや出力データとなり、有効バンク選択部 92 で選択された処理要素が有効に動作し、プログラムデータメモリと処理要素を切り換えながら処理を進めることになる。

#### 【0093】

次に処理装置 70 が複数ある場合の動作について、図 26 の構成を例に説明する。図 26 は図 17 の処理装置 71 を 1 つ含む構成に処理装置 72 を追加して処理装置 70 が 2 つ存在する構成としている。さらに、動作開始のトリガとなるコマンド SQ0A の発行を処理装置 71 ではなく処理装置 72 から実行するようにしている。また、図 22 に示した処理装置 71 のコマンドシーケンス SQ5 を図 27 に示すようなコマンドシーケンス SQ5B に変更し、"halt" の後に "i



interrupt 72, END71”を実行するように追加し、処理装置71の終了を処理装置72へ割り込みで知らせるようにしている。

#### 【0094】

図28は、コマンドSQ0Aの発行を処理装置72が実行し処理装置71が自己の終了を”interrupt”で処理装置72へ通知する上記の動作を示したタイミングチャートである。

#### 【0095】

図28を参照すると、T201において、処理装置72は処理P5の中で、処理装置71の初期化のためのコマンド発行処理SQ0Aを実行する。T201でコマンド発行処理SQ0Aを受けた制御装置60はコマンドシーケンスSQ0を実行する。T202で初期化された処理装置71は図25で示したような一連の処理を行い、T203でコマンド発行処理SQ5Aを実行する。T203で制御装置60はコマンドシーケンスSQ5Bを実行し、”halt”の実行によりT204で処理装置71を停止させ、”interrupt”の実行によりT205で処理装置72に割り込みを出力する。

#### 【0096】

処理装置72は処理P5を実行中に、T201においてコマンド発行処理SQ0Aを実行した後、すぐに別の処理P6を実行することができる。そして、処理P6において処理装置71からの割り込みを待って、後続の処理P7に遷移することもできる。もちろん、処理装置72は処理P6を実行せずに、単に処理装置71からの割り込みを待つだけでもよい。

#### 【0097】

また、図26の構成の場合には、処理装置71で”halt”が実行された後の終了状態を処理装置71の初期状態として、同じ処理を何度でも処理装置72から実行することができる。また、処理要素72は図2、図3、図4、図5で示した構成が可能であり、特にCPU120が処理要素72の構成要素として入っていた場合には、そのソフトウェアプログラムの中の1つの関数呼び出しやシステムコール、もしくはそれらを含むスレッドやオブジェクト、あるいはCPU120のインストラクションの拡張として処理要素71を使用してもよい。

**【0098】**

このように複数の処理装置を並列に動作させる際に、"interrupt"コマンドを利用して処理装置間で同期をとれるので、処理能力を向上させとともに全体の処理の前後関係を順序立てて実行することができる。

**【0099】**

また、電子計算機30は図1に示したように、接続網20を介した外部からコマンドを受けとることができるため、電子計算機30の外部からコマンド発行処理SQ0Aを実行させるようにしてもよい。さらに、図5に示したように、処理装置70の処理要素を電子計算機30で実現することもできるため、処理を階層化・細分化して実装してもよい。

**【0100】**

なお、上記動作説明では、本電子計算機の構成を固定した場合の実装例を示したが、例えば、図5のように制御装置60が再構成可能な制御装置R60であった場合には、図29のような生成フローにおいて、コマンドシーケンス中間コードで使用しているコマンドのみをコマンドセットとして使用するようにし、そのコマンドサブセットのみを実装した制御装置R60を構成してもよい。このように構成することで、制御装置60を簡略化することができる。

**【0101】**

以上の説明では図16に示したキャッシュ装置50を除いた構成で説明したが、図17や図26の構成にキャッシュ装置50を挿入する構成にすることもできることは明瞭であり、キャッシュ装置50を挿入することにより、電子計算機30の外部との間のデータ転送時間を短縮することができ、さらにアドレス変換装置150を有することにより、処理装置に固有のアドレス空間を持たせることが可能になる。

**【0102】**

また、実装フローによる解析結果に基づいて、制御装置60や処理装置70も含めた全ての構成要素を新規に設計および製造してもよい。

**【0103】**

なお、本発明が上記各実施例に限定されず、本発明の技術思想の範囲内におい

て、各実施例は適宜変更され得ることは明らかである。

#### 【0 1 0 4】

##### 【発明の効果】

以上説明したように本発明によれば、アプリケーションプログラムの処理を分割して複数の処理単位とし処理単位毎に論理回路を形成するプログラムを切り換えながら再構成可能ハードウェアで実行させることにより大きなアプリケーションプログラムでも小さな再構成可能ハードウェアで実行可能となるため、アプリケーションプログラムを安価な構成で高速に実行でき、さらに、処理単位のコマンドシーケンスを変更することにより処理単位のプログラムを容易に応用することができるので、新たなアプリケーションプログラムへの応用や、新たな電子計算機の構築を低コストで実現できるという効果がある。

#### 【0 1 0 5】

さらに、論理回路を形成するプログラムを保持するプログラムデータメモリを複数有して処理単位の実行中に次の処理単位のプログラムを別のプログラムデータメモリに読み出すことにより、切り換えの際のプログラムの読み出し時間を短縮して切り換え時間を短縮し処理速度を向上することができる。

##### 【図面の簡単な説明】

##### 【図 1】

本発明の実施形態の構成を示すブロック図である。

##### 【図 2】

本発明の実施形態の処理装置の一例を示すブロック図である。

##### 【図 3】

本発明の実施形態の処理要素の一例を示した図である。

##### 【図 4】

本発明の実施形態の処理要素の一例を示した図である。

##### 【図 5】

本発明の実施形態の処理要素の一例を示した図である。

##### 【図 6】

本発明の実施形態の制御装置と処理要素の接続例を示した図である。

**【図 7】**

本発明の実施形態の制御装置に実装するコマンドコードの構成を示した図である。

**【図 8】**

本発明の実施形態の制御装置に実装するコマンドコードの機能を示した図である。

**【図 9】**

本発明の実施形態の制御装置と処理要素の接続例を示した図である。

**【図 10】**

本発明の実施形態のコマンドの格納例を示した図である。

**【図 11】**

本発明の実施形態の制御装置と処理要素の接続例を示した図である。

**【図 12】**

本発明の実施形態のコマンドの格納例を示した図である。

**【図 13】**

本発明の実施形態のアドレスカウンタ制御コードの構成例を示した図である。

**【図 14】**

本発明の実施形態のアドレスカウンタ制御コードの機能例を示した図である。

**【図 15】**

本発明の実施形態のコマンドシーケンスを示した図である。

**【図 16】**

本発明の実施形態のキャッシュを付加した構成のブロック図である。

**【図 17】**

本発明の実施形態の動作説明で参照する構成例を示すブロック図である。

**【図 18】**

本発明の実施形態の実装するアプリケーションを示した図である。

**【図 19】**

本発明の実施形態のプログラムデータの生成フロー図である。

**【図 20】**

本発明の実施形態のコマンドシーケンス中間コードを示した図である。

【図 2 1】

本発明の実施形態の処理の制御フロー図である。

【図 2 2】

本発明の実施形態のコマンドコードメモリに割り当てたコマンドシーケンスの図である。

【図 2 3】

本発明の実施形態のメモリに割り当てたプログラムデータの図である。

【図 2 4】

本発明の実施形態の電源投入時の電子計算機の状態を示した図である。

【図 2 5】

本発明の実施形態の動作を示したタイミングチャートである。

【図 2 6】

本発明の実施形態の 2 つの処理装置で構成するブロック図である。

【図 2 7】

本発明の実施形態のメモリに割り当てたコマンドシーケンスの図である。

【図 2 8】

本発明の実施形態の 2 つの処理装置が動作するタイミングチャートである。

【図 2 9】

本発明の実施形態のプログラムデータの生成フロー図である。

【図 3 0】

従来技術においてアプリケーションプログラムの全てを専用ハードウェアとして実装する手法例を示した図である。

【図 3 1】

従来技術においてアプリケーションプログラムの一部を専用ハードウェアとして実装する手法例を示した図である。

【図 3 2】

従来技術においてアプリケーションプログラムの一部を専用ハードウェアとして実装する手法例を示した図である。

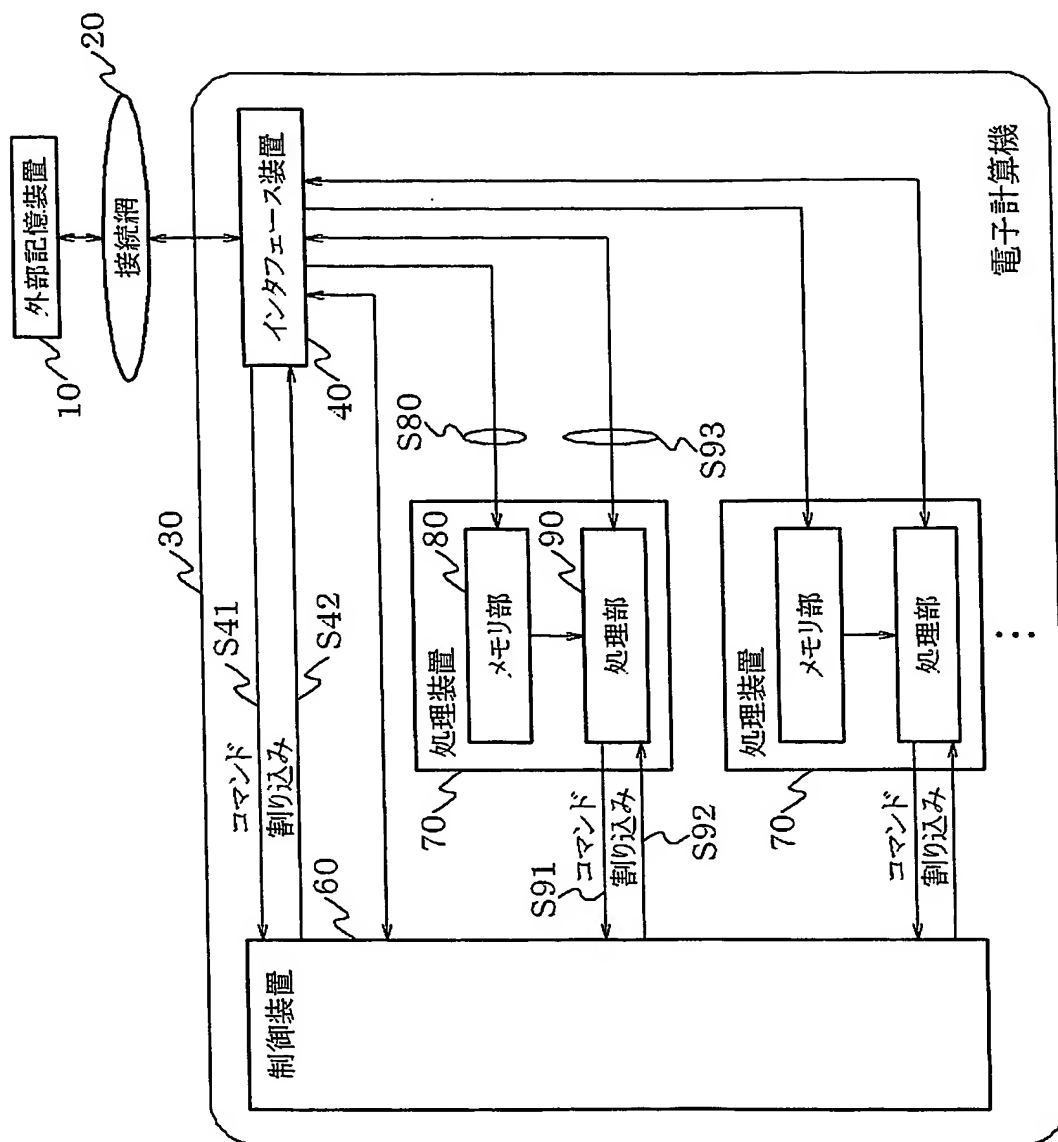
## 【符号の説明】

- 1 0 外部記憶装置
- 2 0 接続網
- 3 0 電子計算機
- 4 0 インタフェース装置
- 5 0 キャッシュ装置
- 6 0 制御装置
- 6 1 コマンドコード参照装置
- 6 2 アドレスカウンタ
- 6 3 コマンドコードメモリ
- 7 0、7 1、7 2 処理装置
- 8 0 メモリ部
- 8 1、8 1 1、8 1 2、8 1 3 プログラムデータメモリ
- 8 2 有効ブロック選択部
- 8 3 選択メモリ
- 9 0 処理部
- 9 1 処理要素
- 9 2 有効バンク選択部
- 1 0 1、1 0 2 バンク
- 1 1 0 処理要素
- 1 2 0 C P U
- 1 3 0 キャッシュコントローラ
- 1 4 0 キャッシュメモリ
- 1 5 0、1 5 1 アドレス変換装置
- R 6 0 再構成可能制御装置
- M 1 制御フロー解析手順
- M 2 コマンドシーケンス実装手順
- M 3 プログラムデータ生成手順

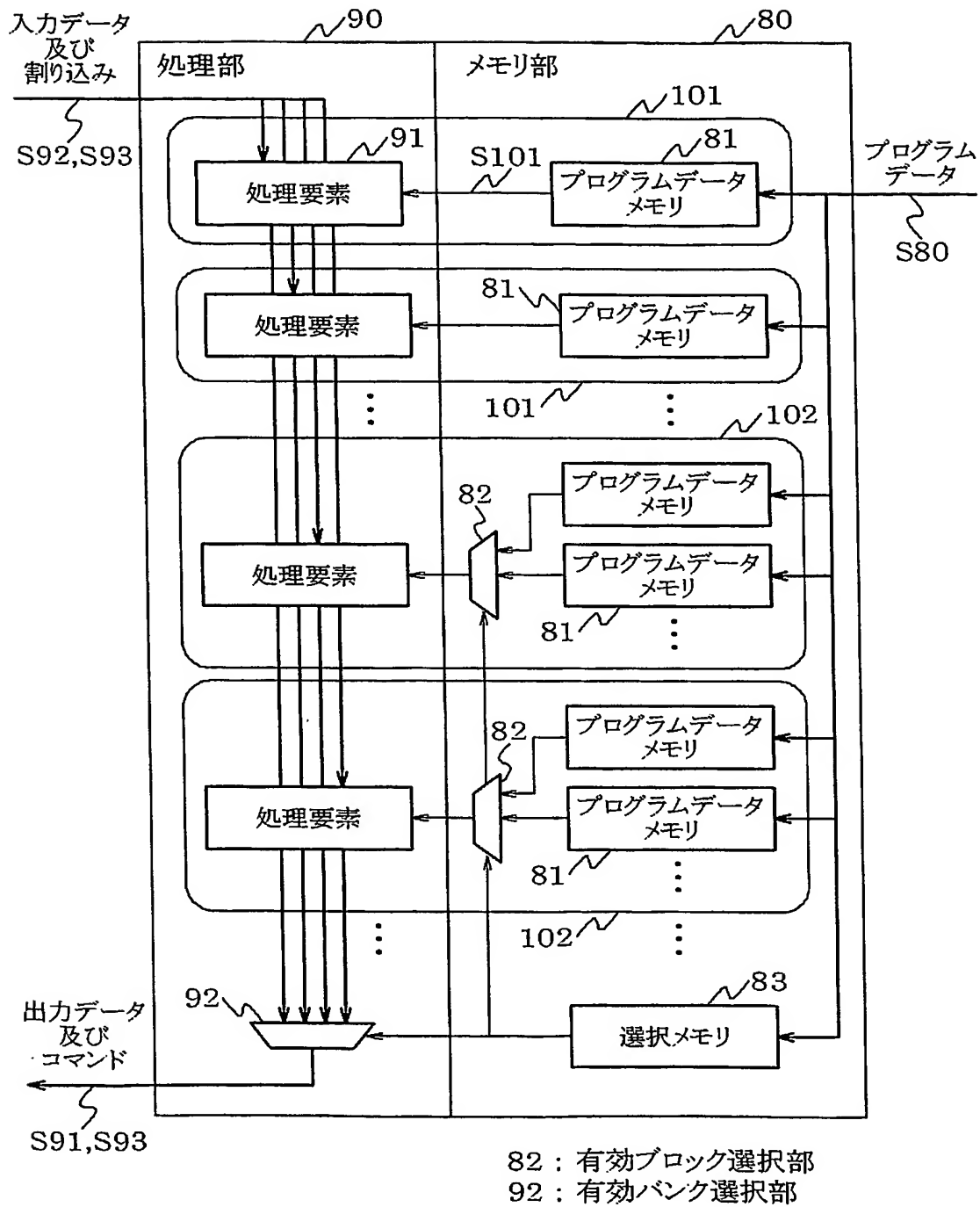
【書類名】

図面

【図 1】

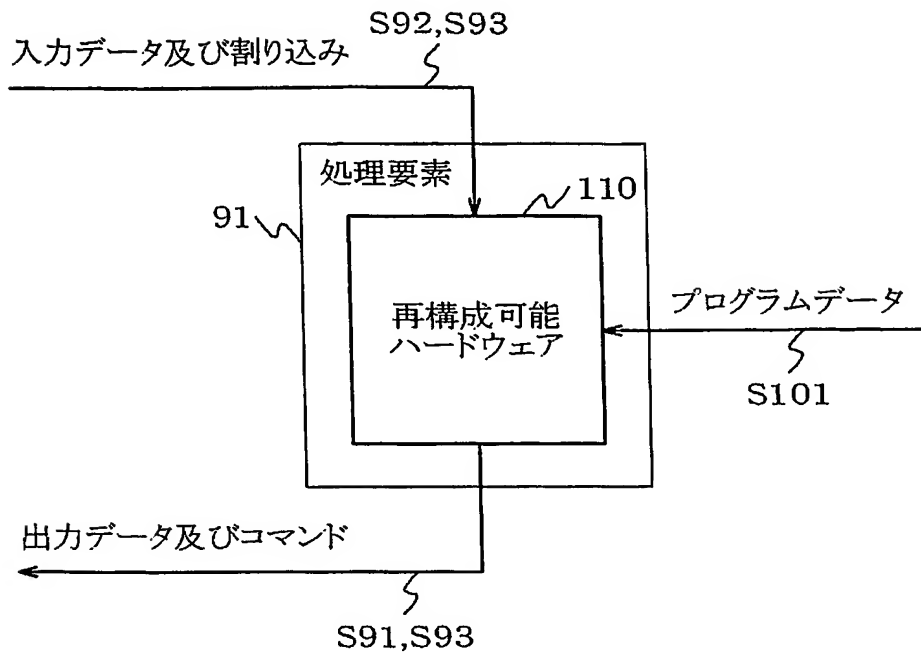


【図 2】

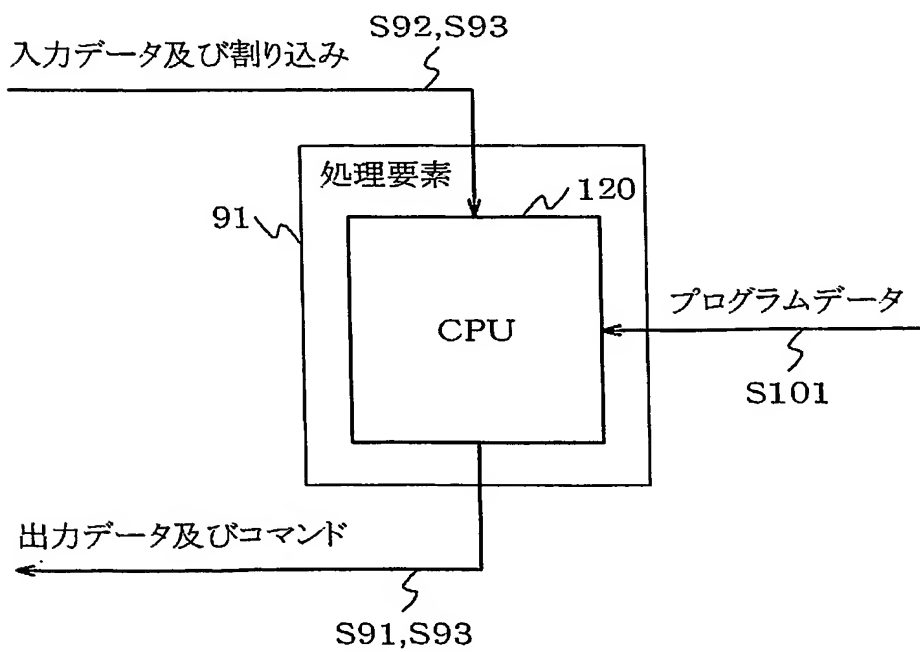




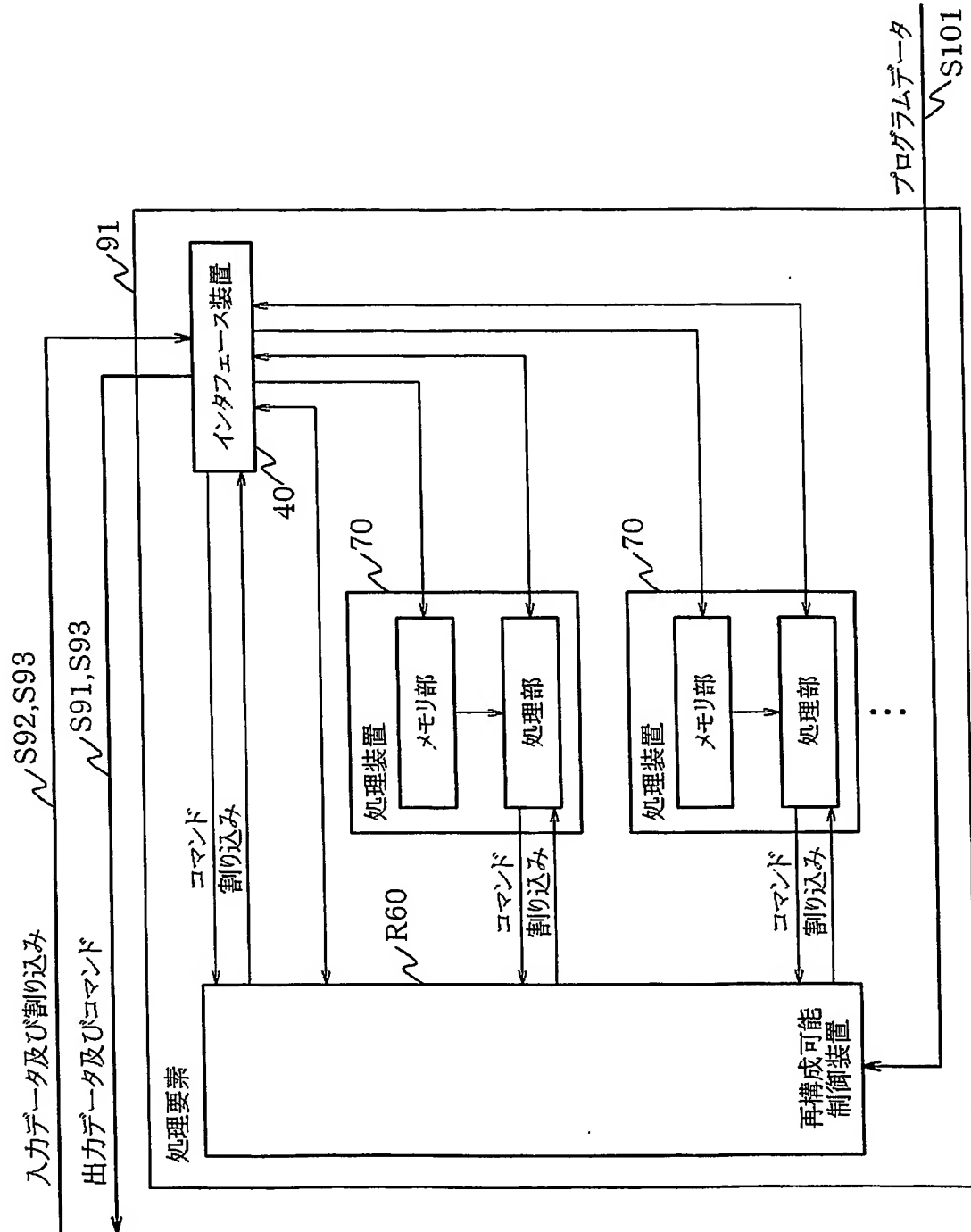
【図 3】



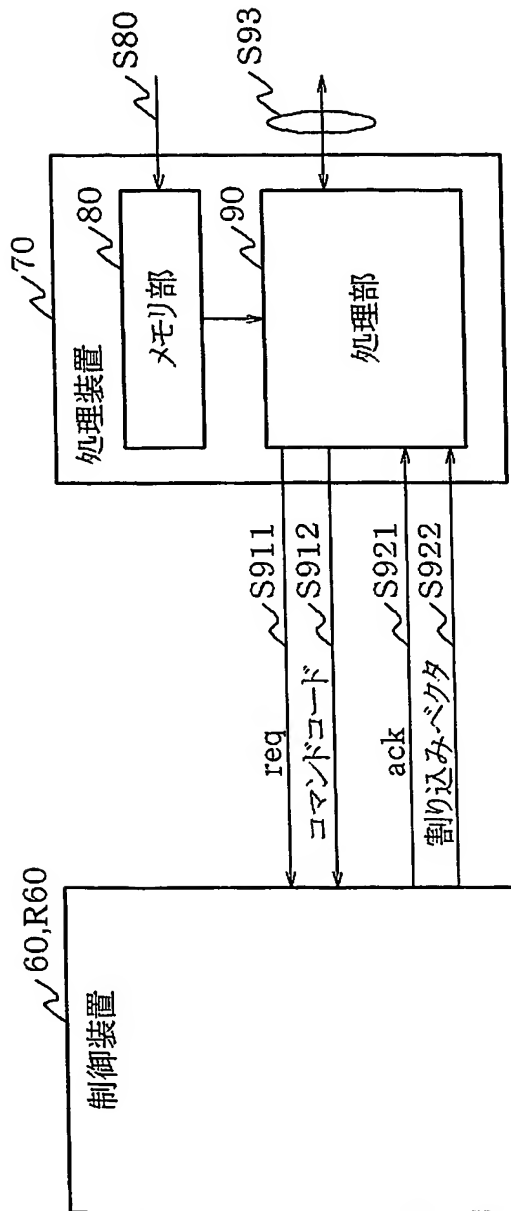
【図 4】



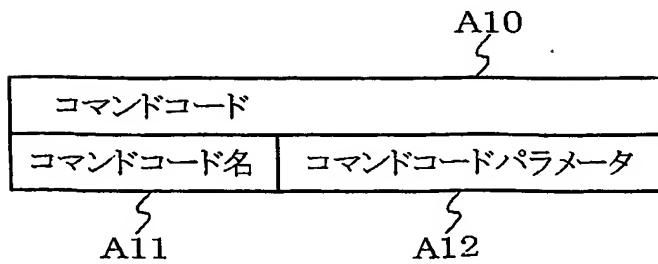
【図 5】



【図 6】



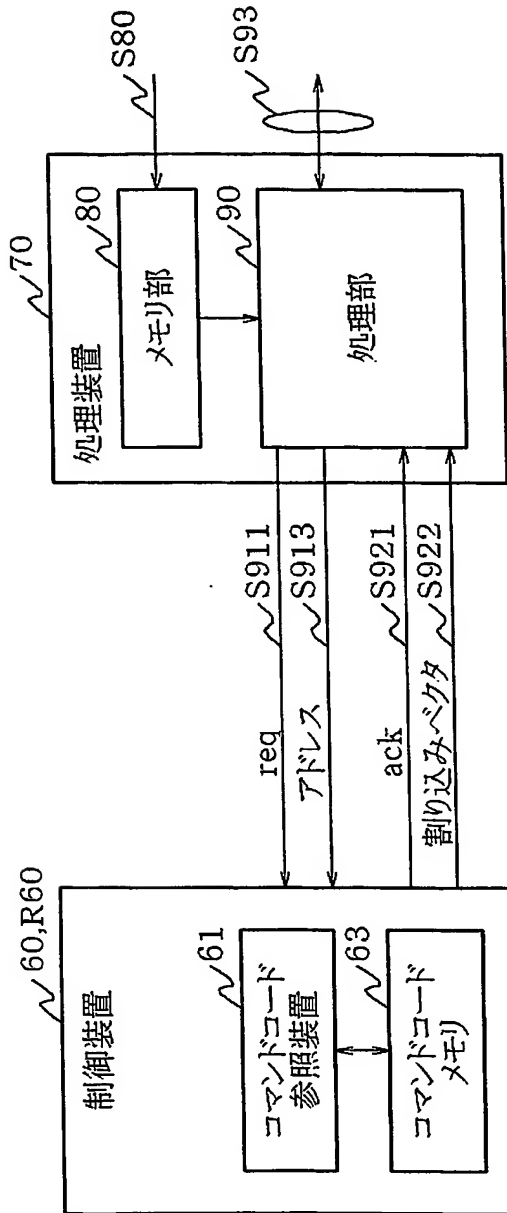
【図 7】



【図 8】

コマンドコード名	コマンドコードパラメータ	内容
activate	処理要素	指定した処理要素を選択し、動作を開始させる
halt	処理装置	指定した処理装置の動作を停止する
Interrupt	処理装置、及び割り込みベクタ番号	指定した処理装置に指定した割り込みベクタ番号を発行する
load_prg	プログラムデータメモリの領域、及び プログラムデータが格納されているメモリ領域	指定したプログラムデータメモリの領域に指定したメモリ領域に格納されているプログラムデータを転送する
cancel_prg	プログラムデータメモリの領域、及び プログラムデータが格納されているメモリ領域	指定したプログラムデータメモリの領域に指定したメモリ領域に格納されているプログラムデータを転送することを中止する
wait_prg	プログラムデータメモリの領域、及び プログラムデータが格納されているメモリ領域	指定したプログラムデータメモリの領域に指定したメモリ領域に格納されているプログラムデータが全て転送されるまで待機する

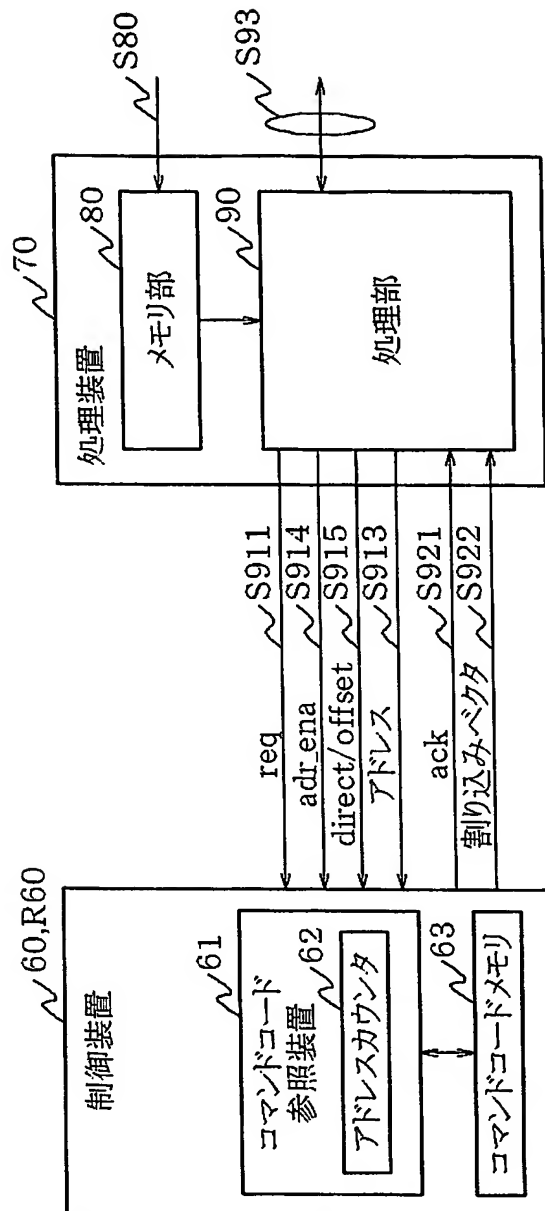
【図 9】



【図 10】

アドレス	データ
アドレス1	コマンドコード1
アドレス2	コマンドコード2
⋮	⋮
	A10

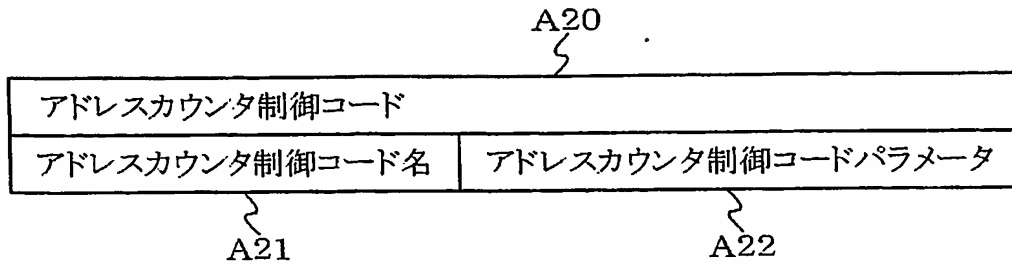
【図 11】



【図 12】

アドレス	データ		
アドレス1	アドレスカウンタ制御コード1	フラグ1	コマンドコード1
アドレス2	アドレスカウンタ制御コード2	フラグ2	コマンドコード2
	A20	⋮	A30
			A10

【図 13】









【図 14】

アドレスカウンタ 制御コード名	アドレスカウンタ 制御コードパラメータ	内容
load_adr	N	アドレスカウンタの値にNを設定する
add_adr	N	アドレスカウンタの値にNを加算する
push_adr	N	アドレスカウンタの値をアドレスカウンタスタックに退避し、Nに設定する
pop_adr		アドレスカウンタの値をアドレスカウンタスタックから復帰させる

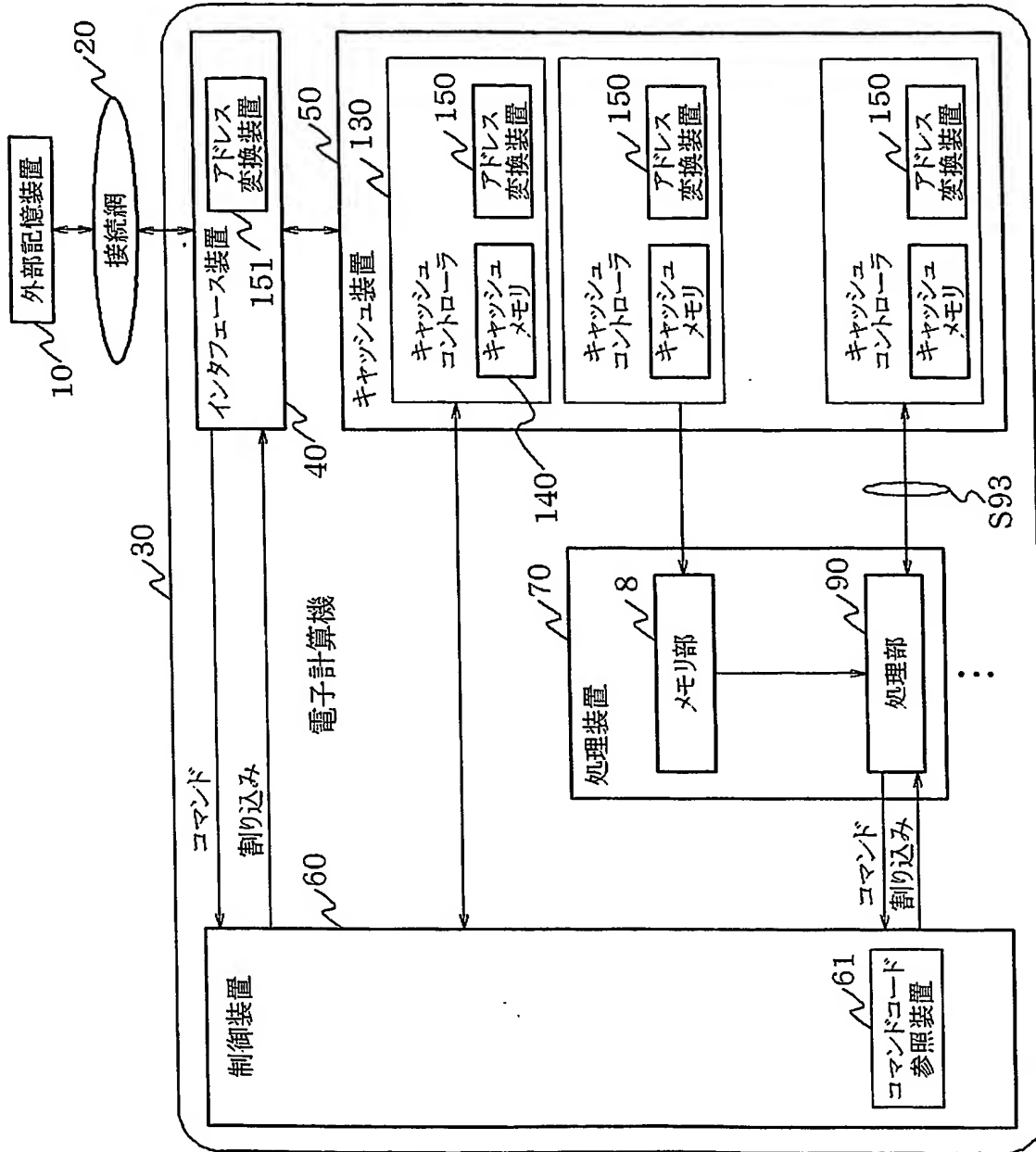
N: 数値



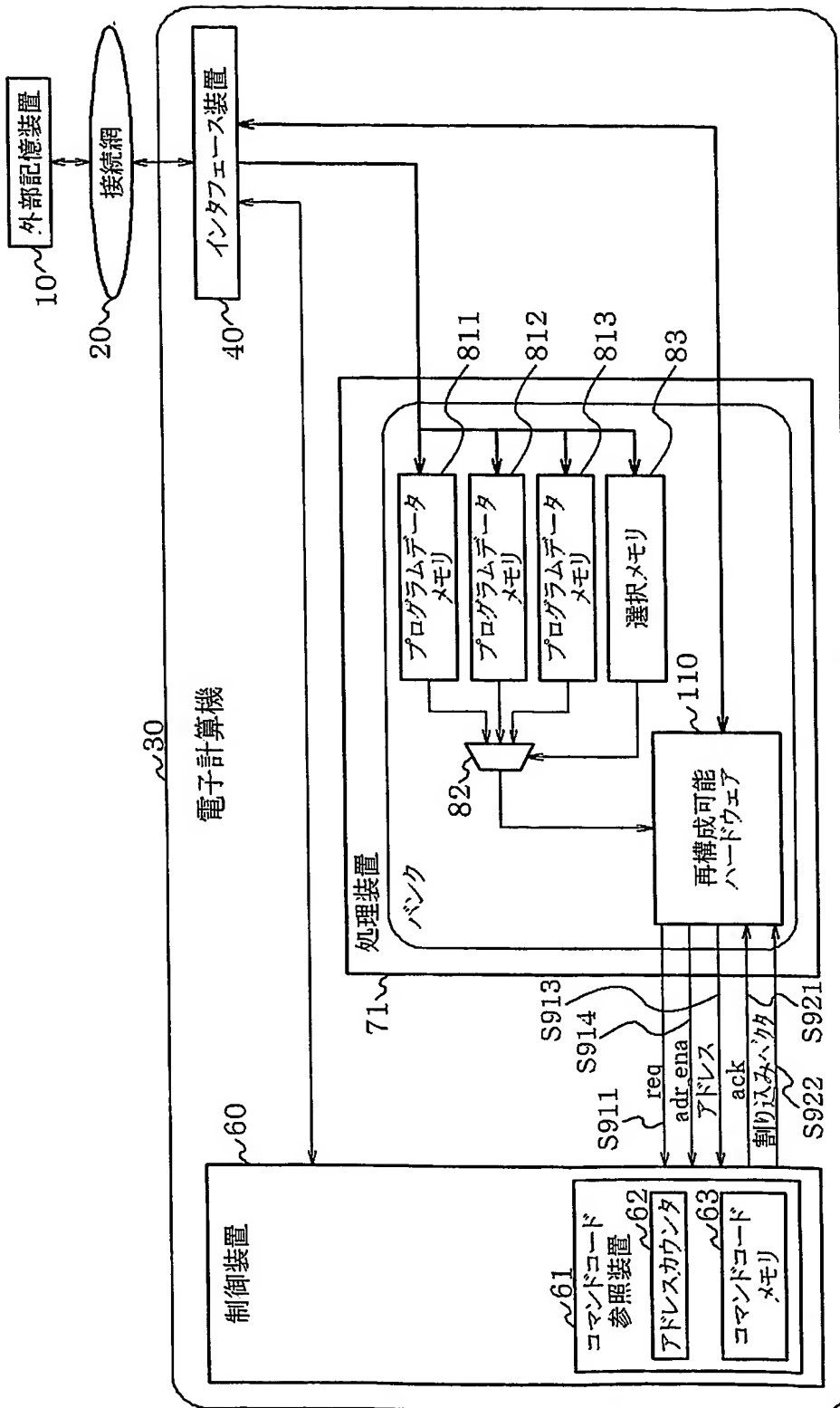
【図 15】

アドレス	データ	
	アドレスカウンタ制御コード及びフラグ	コマンドコード
⋮	 X100                     ⋮  Y100	
100	add_adr 1   cont	コマンドコード100
101	load_adr 200   cont	コマンドコード101
⋮	 X200                     ⋮  X101  Y101                     Y200 	
200	add_adr 1   stop	コマンドコード100
201	アドレスカウンタ制御コード201	コマンドコード101
⋮	⋮	

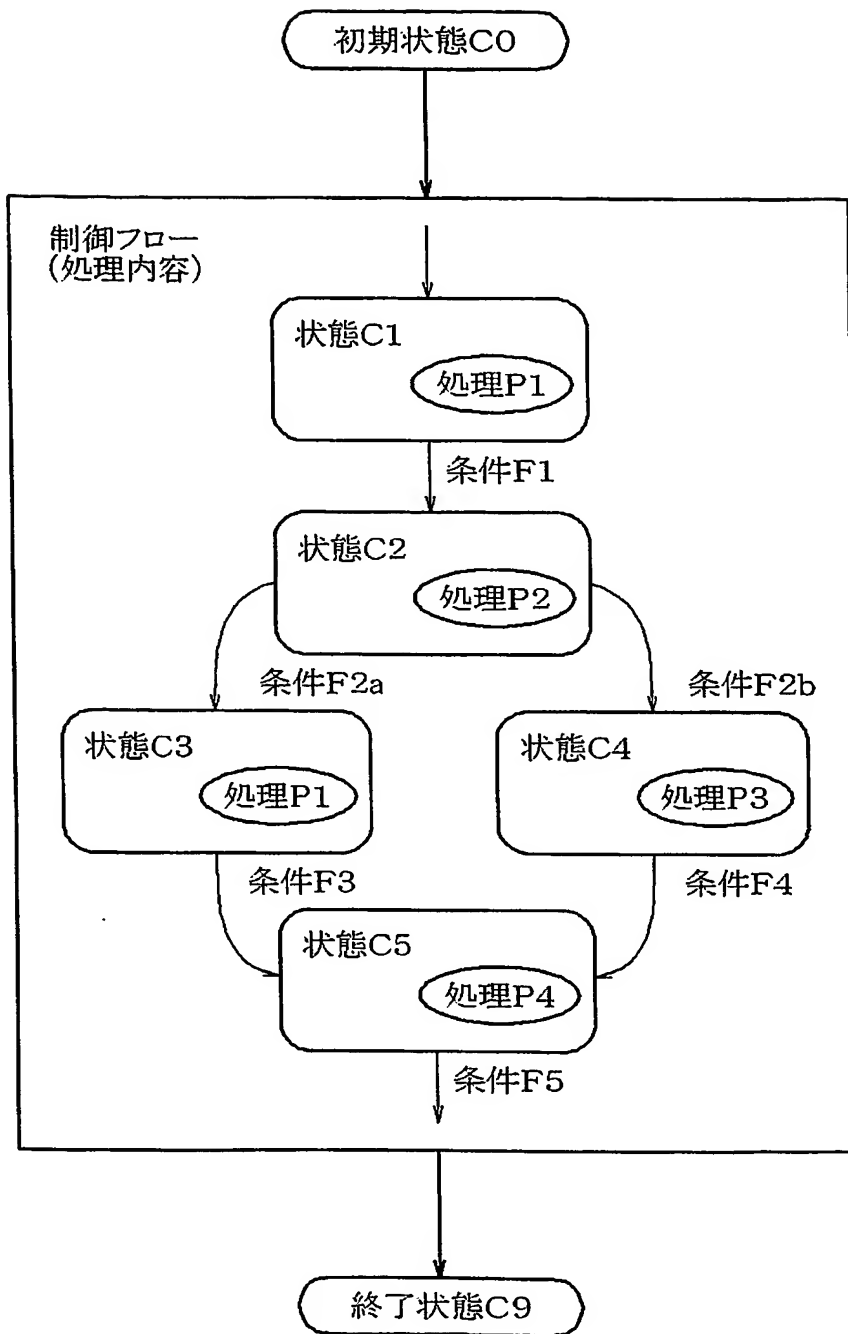
【図 16】



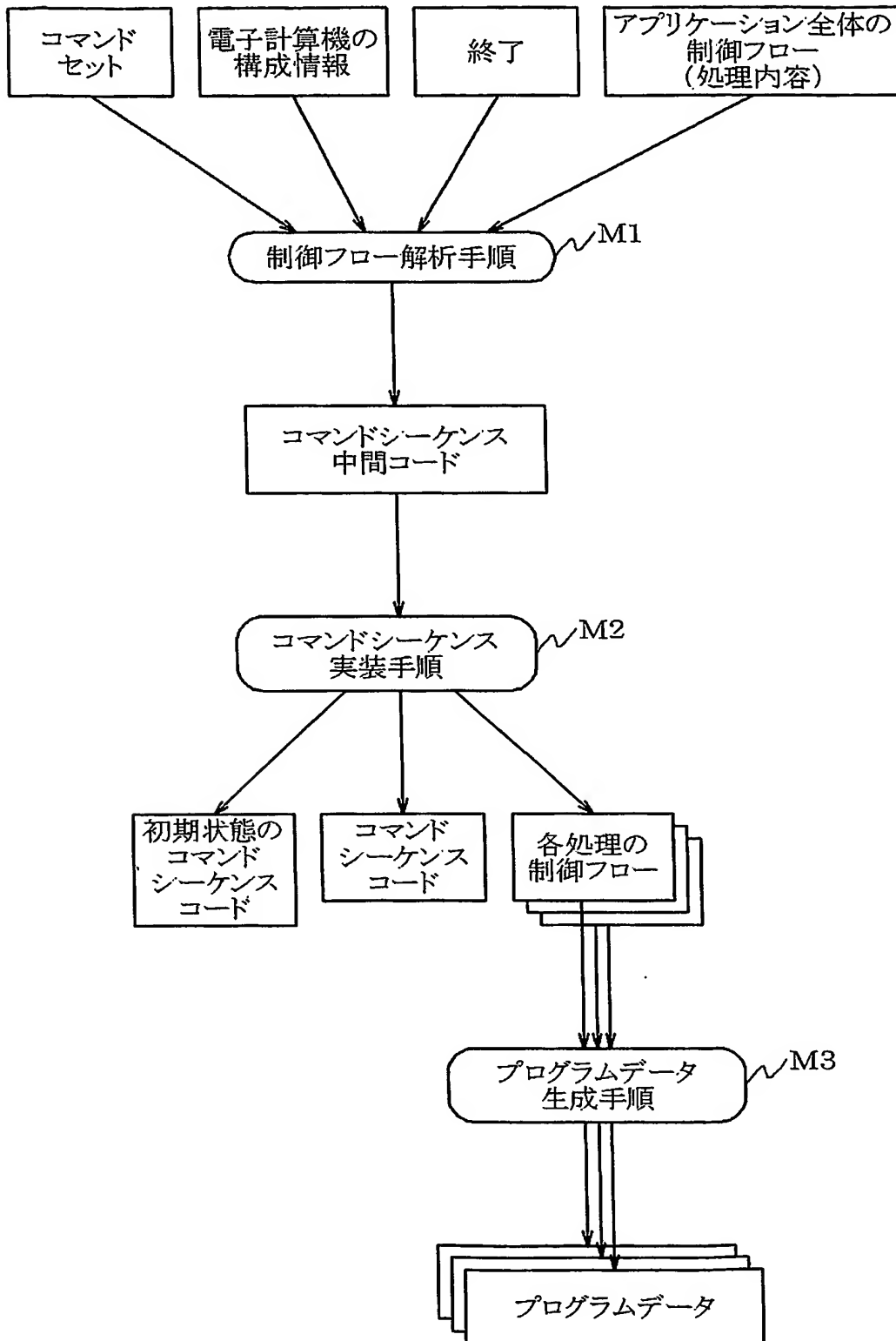
【図17】



【図 18】



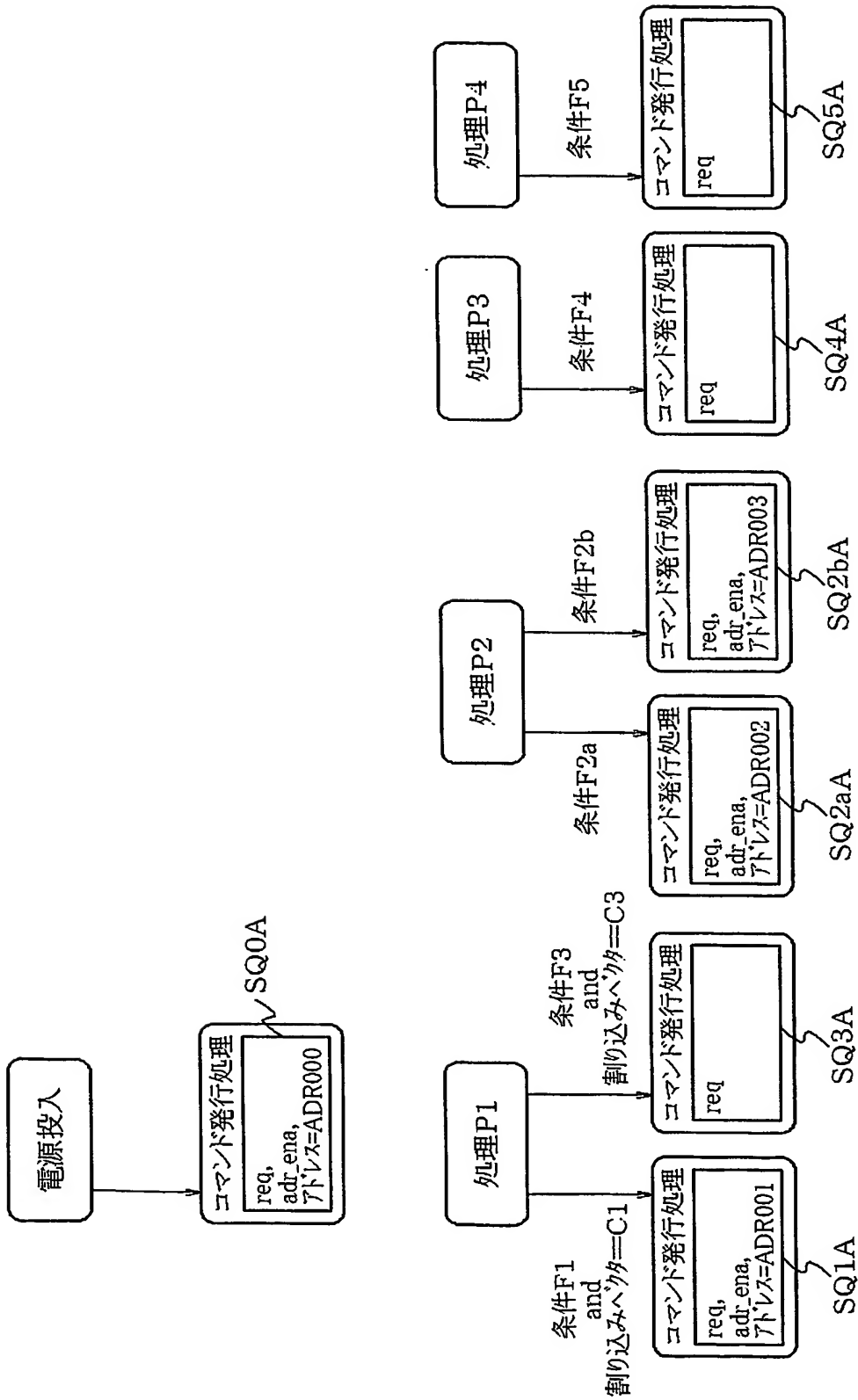
【図19】



【図20】

処理内容	状態	遷移条件	遷移先	使用する プログラム データメモリ	コマンドシーケンス 中間コード
—	初期状態 C0	—	状態 C1	811	load_prg 812, PM1 wait_prg 812, PM1 interrupt 71, C1 activate 812 load_prg 813, PM2
	状態 C1		状態 C2		wait_prg 813, PM2 activate 813 load_prg 811, PM3
処理 P1	状態 C3	条件F3 and 割り込みベクタ==C3	状態 C5	812	wait_prg 813, PM4 activate 813
	状態 C2	条件 F2a	状態 C3		cancel_prg 811, PM3 interrupt 71, C3 activate 812 load_prg 813, PM4
処理 P2	状態 C4	条件 F2b	状態 C4	813	wait_prg 811, PM3 activate 811 load_prg 813, PM4
		条件 F4	状態 C5		wait_prg 813, PM4 activate 813
処理 P3	状態 C5	条件 F5	終了状態 C9	811	halt 71

【図 21】



【図 22】

ベースアドレス値	オフセット値	コマンドシケンス	
		アドレスカウンタ制御コード	コマンドコード
ADR000	+0	add_adr 1	load_prg 812, PM1
	+1	add_adr 1	wait_prg 812, PM1
	+2	add_adr 1	interrupt 71, C1
	+3	add_adr 1	activate 812
	+4	add_adr 0	load_prg 813, PM2
ADR001	+0	add_adr 1	wait_prg 813, PM2
	+1	add_adr 1	activate 813
	+2	add_adr 0	load_prg 811, PM3
ADR002	+0	add_adr 1	cancel_prg 811, PM3
	+1	add_adr 1	interrupt 71, C3
	+2	add_adr 1	activate 812
	+3	load_adr ADR004	load_prg 813, PM4
ADR003	+0	add_adr 1	wait_prg 811, PM3
	+1	add_adr 1	activate 811
	+2	load_adr ADR004	load_prg 813, PM4
ADR004	+0	add_adr 1	wait_prg 813, PM4
	+1	add_adr 1	activate 813
	+2	add_adr 0	halt 71

SQ0

SQ1

SQ2a

SQ2b

SQ3, SQ4

SQ5



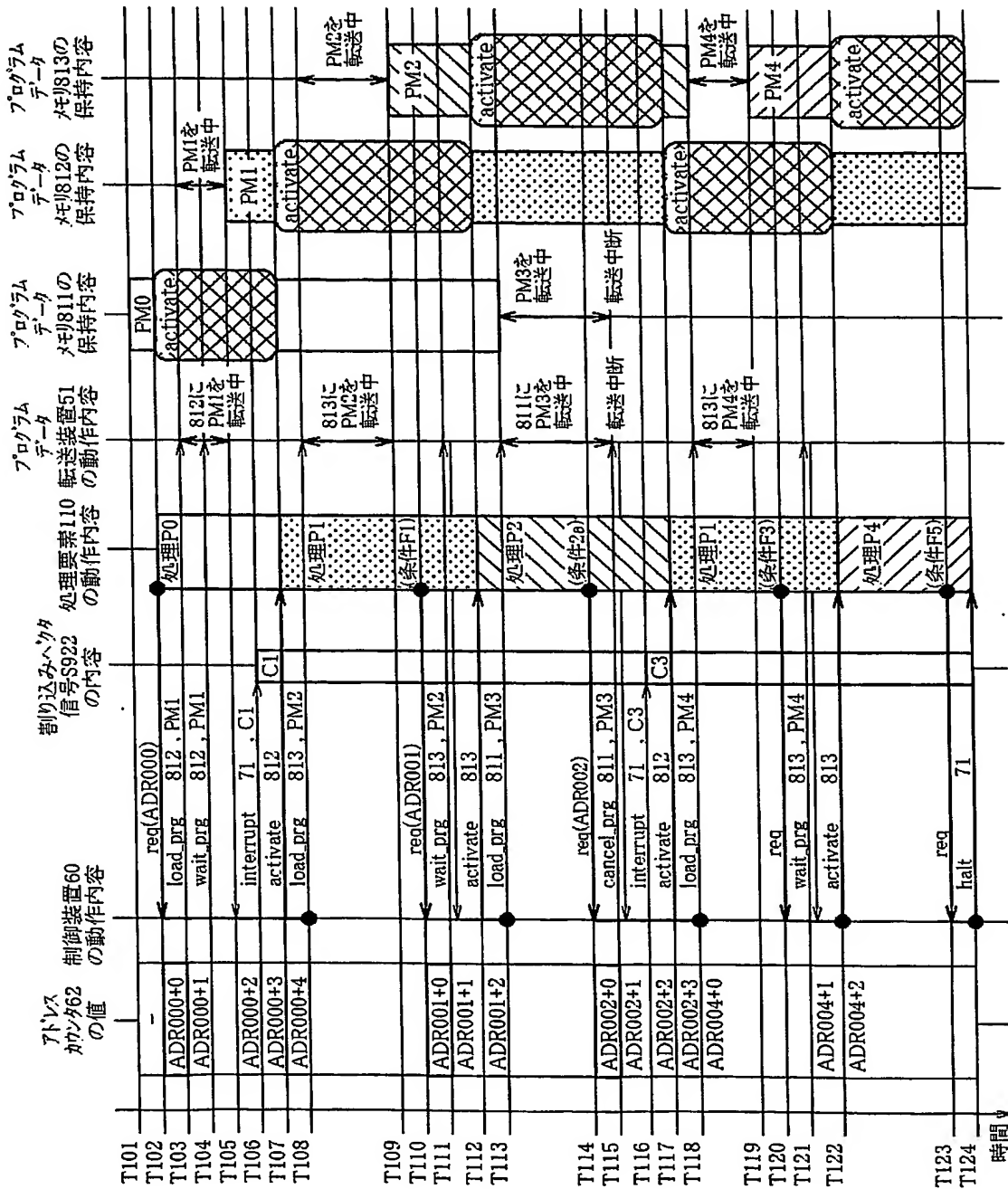
【図 2 3】

メモリ	
SQ0Aのコマンド発行 を行なうプログラムデータ	PM0
処理P1、SQ1A、SQ3Aのコマンド発行 を行なうプログラムデータ	PM1
処理P2、SQ2aA、SQ2bAのコマンド発行 を行なうプログラムデータ	PM2
処理P3、SQ4Aのコマンド発行 を行なうプログラムデータ	PM3
処理P4、SQ5Aのコマンド発行 を行なうプログラムデータ	PM4

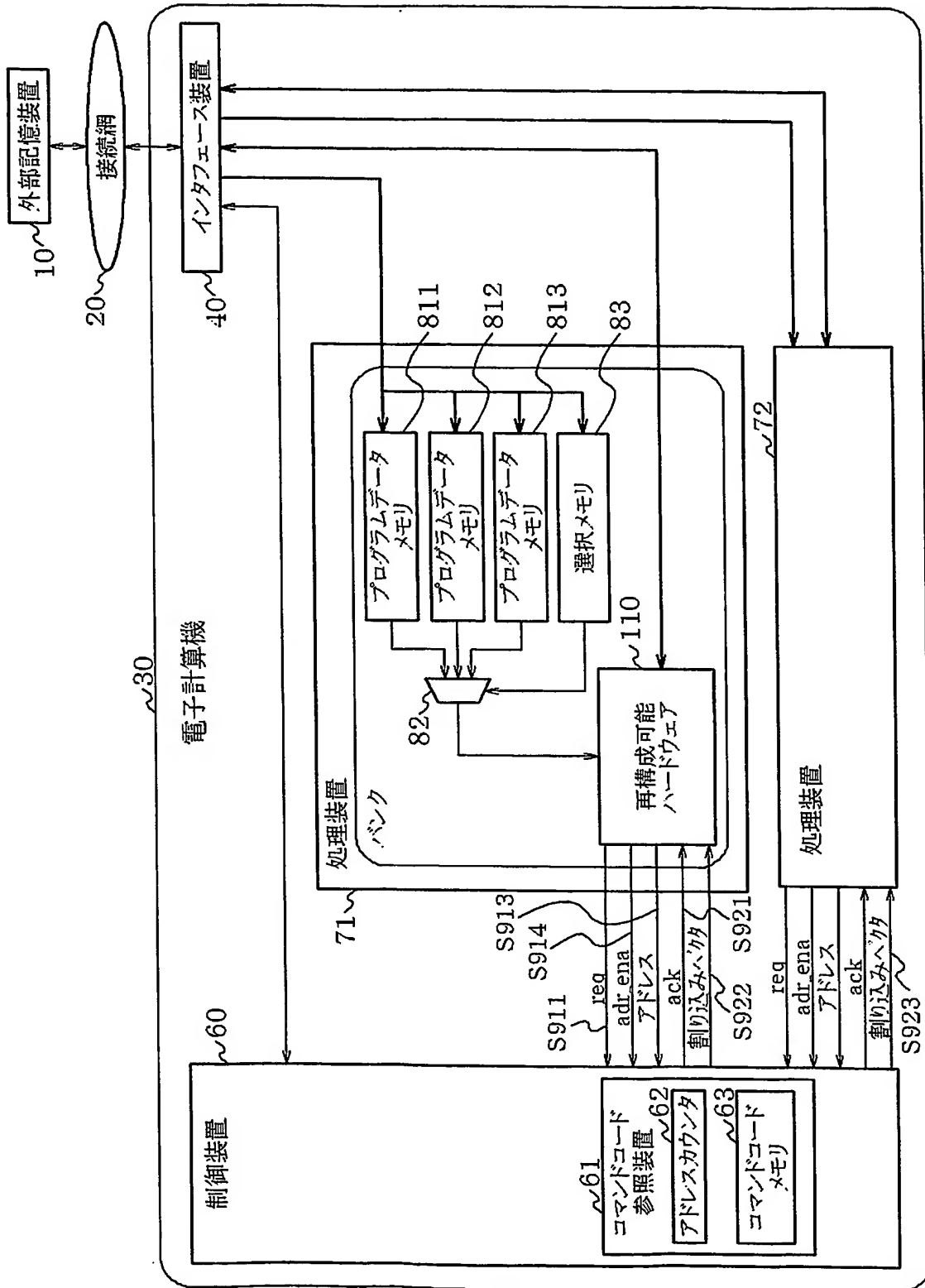
【図 2 4】

プログラムデータメモリ811にPM0が格納されている  
プログラムデータメモリ811がactivateされている

【図 25】



【図26】



【図 27】

ベースアドレス値	オフセット値	コマンドシーケンス		
		アドレスカウンタ制御コード	コマンドコード	
ADR000	+0	add_adr 1	cont	load_prg 812, PM1
	+1	add_adr 1	cont	wait_prg 812, PM1
	+2	add_adr 1	cont	interrupt 71, C1
	+3	add_adr 1	cont	activate 812
	+4	add_adr 0	stop	load_prg 813, PM2
ADR001	+0	add_adr 1	cont	wait_prg 813, PM2
	+1	add_adr 1	cont	activate 813
	+2	add_adr 0	stop	load_prg 811, PM3
ADR002	+0	add_adr 1	cont	cancel_prg 811, PM3
	+1	add_adr 1	cont	interrupt 71, C3
	+2	add_adr 1	cont	activate 812
	+3	load_adr ADR004	stop	load_prg 813, PM4
ADR003	+0	add_adr 1	cont	wait_prg 811, PM3
	+1	add_adr 1	cont	activate 811
	+2	load_adr ADR004	stop	load_prg 813, PM4
ADR004	+0	add_adr 1	cont	wait_prg 813, PM4
	+1	add_adr 1	stop	activate 813
	+2	add_adr 1	cont	halt 71
	+3	add_adr 0	stop	interrupt 72,END71

✓ SQ0

✓ SQ1

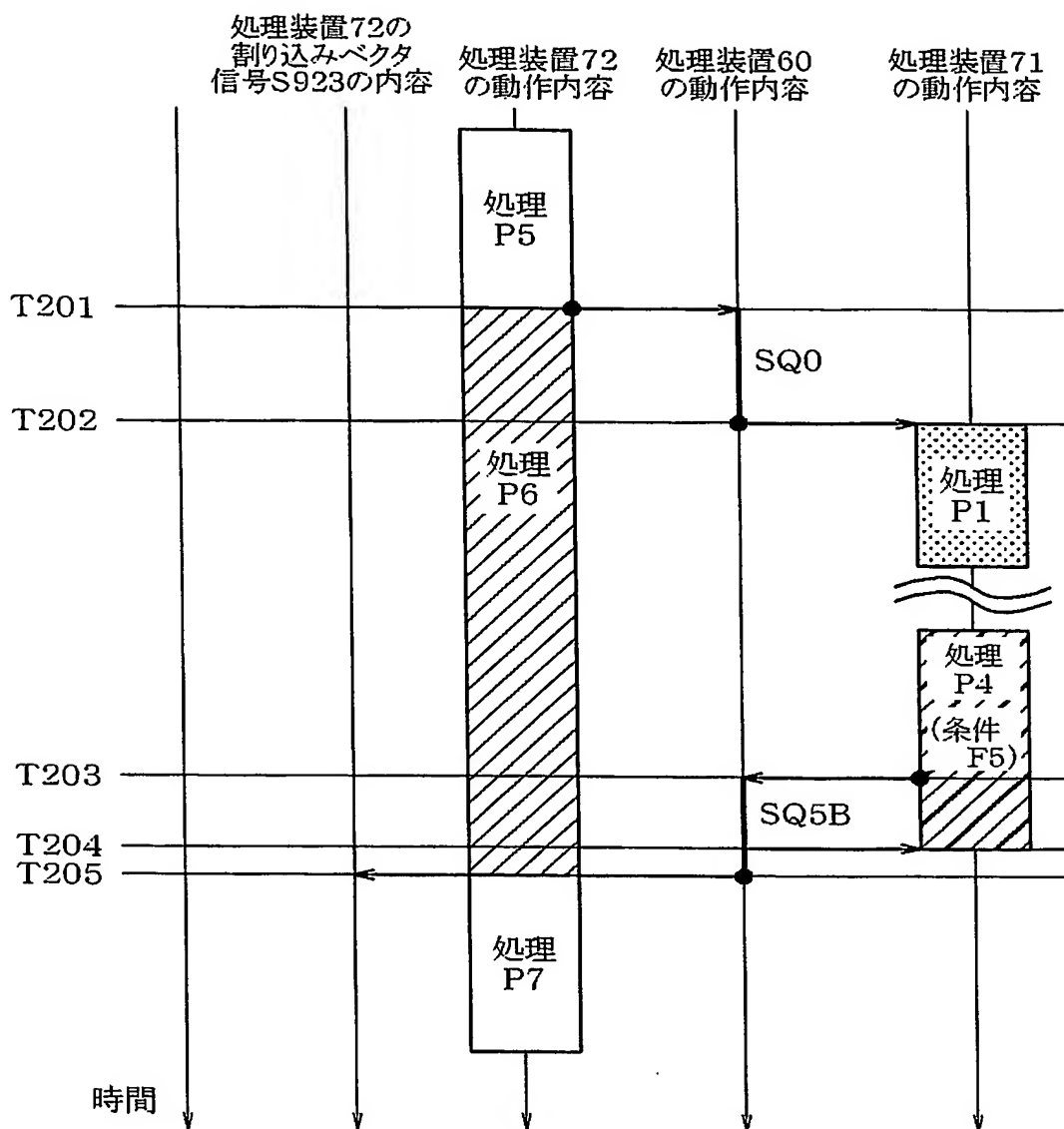
✓ SQ2a

✓ SQ2b

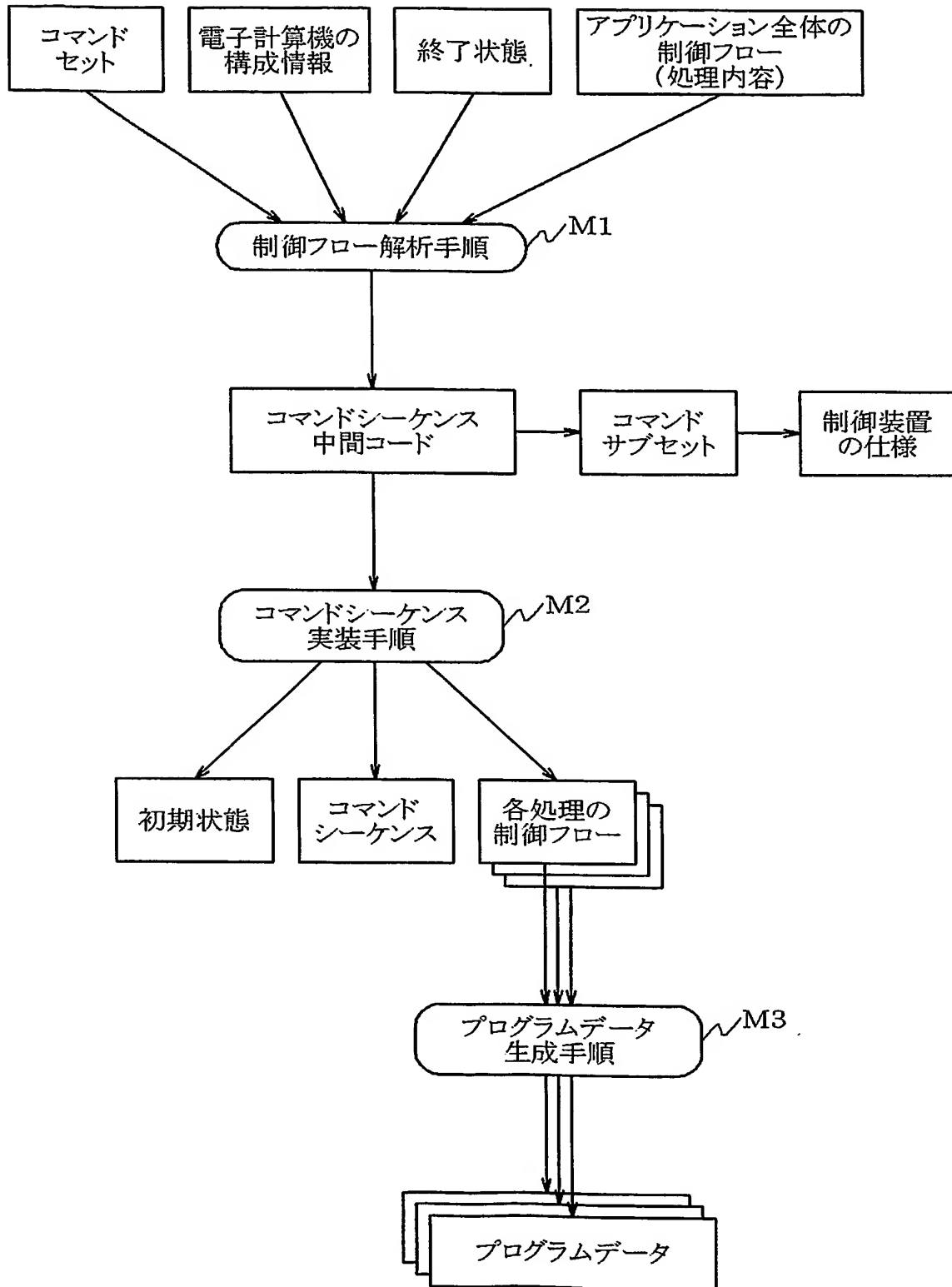
✓ SQ3,SQ4

✓ SQ5B

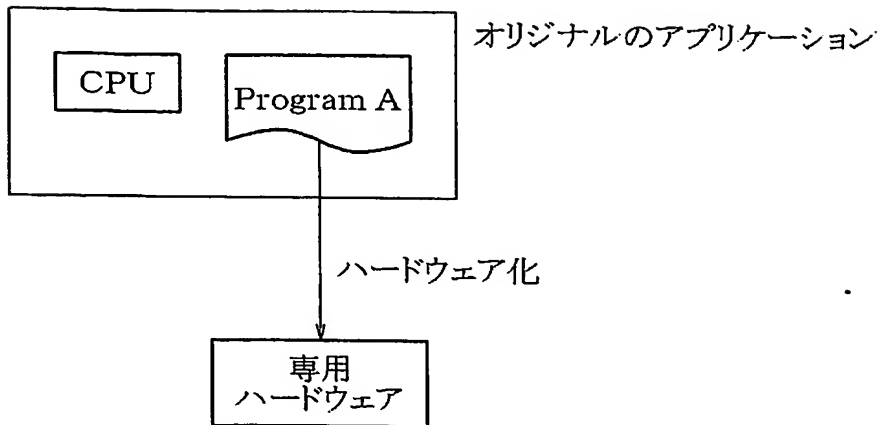
【図28】



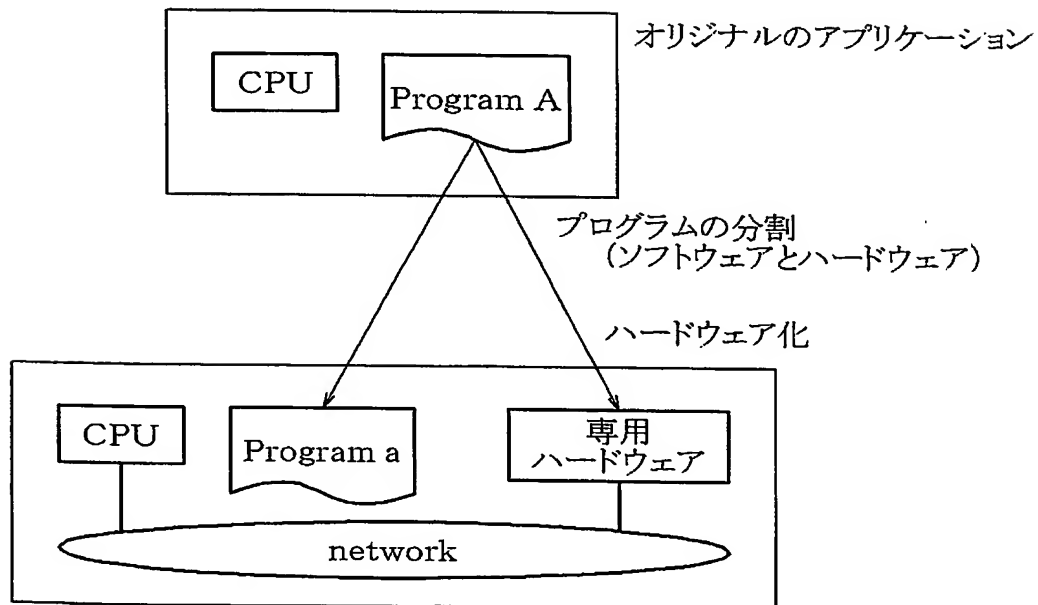
【図 29】



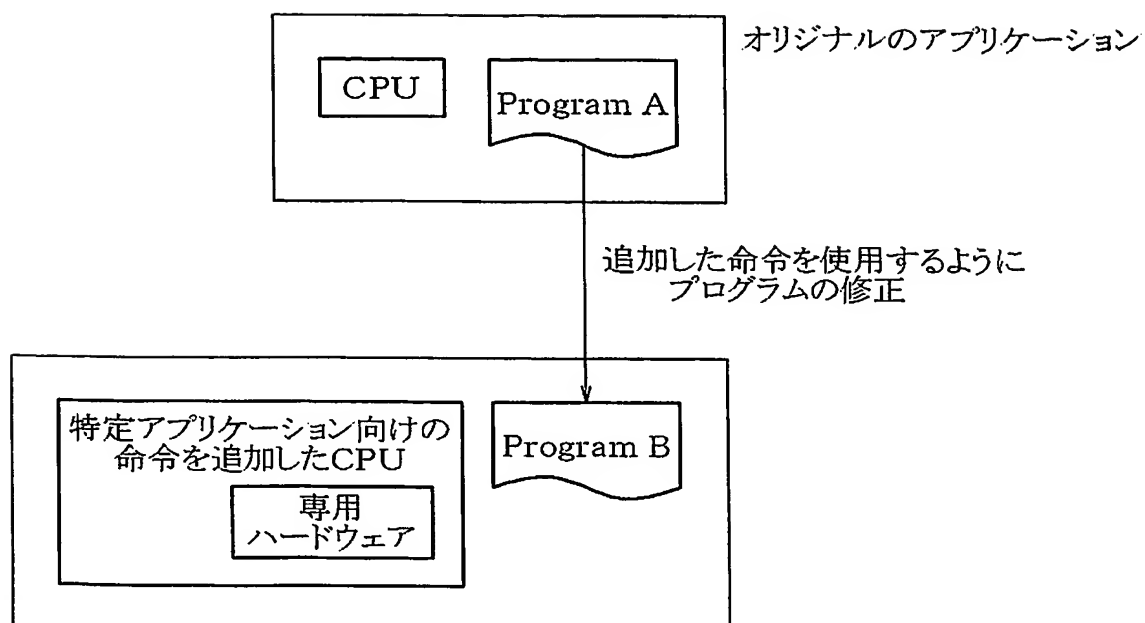
【図 30】



【図 31】



【図 3 2】





【書類名】 要約書

【要約】

【課題】 アプリケーションプログラムを処理単位に分割して切り換えて再構成可能ハードウェアに論理回路を構築して実行させることにより低コストで処理速度を向上させ、アプリケーションプログラムを容易に再利用可能とすること。

【解決手段】 論理回路をプログラムにより形成可能な再構成可能ハードウェアをからなる処理部 90 と処理部 90 の論理回路の形成を決めるプログラムを保持するメモリ部 80 を含む処理装置 70 と、処理装置 70 の指定したコマンドを実行する制御装置 60 とを有し、前記コマンドは、処理装置 70 が所定の条件を検出した際に処理部 90 から実行を指示され、前記メモリ部 80 に保持するプログラムを置き換えたりメモリ部 80 が複数のプログラムデータメモリ 81 から構成される場合に有効とするプログラムデータメモリを切り換えたりするコマンドを含む。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2003-180659
受付番号	50301056065
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年 6月26日

<認定情報・付加情報>

【提出日】	平成15年 6月25日
-------	-------------

次頁無

特願 2 0 0 3 - 1 8 0 6 5 9

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 4 2 3 7 ]

1. 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

東京都港区芝五丁目 7 番 1 号

氏 名

日本電気株式会社

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☒ FADED TEXT OR DRAWING

☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☒ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**